

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-289545

(P2003-289545A)

(43) 公開日 平成15年10月10日 (2003. 10. 10)

(51) Int.Cl. ⁷	識別記号	F I	テームト [*] (参考)
H 0 4 N 7/32		H 0 4 N 7/137	Z 5 B 0 1 5
G 1 1 C 11/41		G 1 1 C 11/34	K 5 C 0 5 9

審査請求 未請求 請求項の数 6 O L (全 32 頁)

(21) 出願番号 特願2002-378128 (P2002-378128)

(22) 出願日 平成14年12月26日 (2002. 12. 26)

(31) 優先権主張番号 特願2002-4958 (P2002-4958)

(32) 優先日 平成14年1月11日 (2002. 1. 11)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 近藤 哲二郎

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72) 発明者 市川 勉

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(74) 代理人 100090376

弁理士 山口 邦夫 (外1名)

最終頁に続く

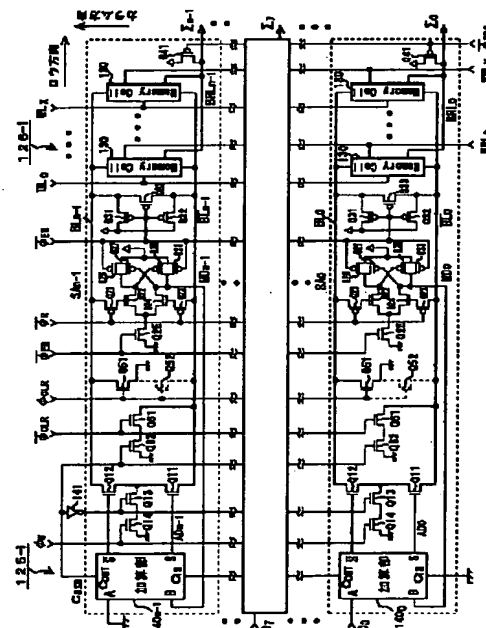
(54) 【発明の名称】 半導体装置

(57) 【要約】

【課題】 演算器とメモリとの間のデータのやり取りを効率よく行い得るようにする。

【解決手段】 記憶領域126_{n-1}のカラム方向に延びる各行のn個のメモリセル130は、夫々1つの記憶部を構成する。メモリ領域126_{n-1}のロウ方向に並ぶ各列のメモリセル130に夫々対応して、センスアンプSA₀~SA_{n-1}を配している。加算器125_{n-1}はnビットのそれぞれのビットの加算を行うためのn個の加算部140₀~140_{n-1}からなり、これらn個の加算部140₀~140_{n-1}はメモリ領域126_{n-1}のカラムのビットに描えて配されている。加算部140₀~140_{n-1}に所定の記憶部のnビットのメモリセル130の記憶データをセンスアンプSA₀~SA_{n-1}を介して供給し、これに入力データD₀~D_nを足し込むことができる。また、足し込みデータをビット線を通じて上述の所定の記憶部のnビットのメモリセル130に書き戻すことができる。

半導体メモリと加算器とを一体化した構成



【特許請求の範囲】

【請求項1】 演算器およびメモリが一体化されてなり、
上記演算器を構成するビット単位の複数の演算部は、それぞれ上記メモリのカラムのピッチに揃えて配されていることを特徴とする半導体装置。

【請求項2】 上記演算器は加算器であり、
入力データと上記メモリの記憶データとを上記複数の演算部で加算し、該加算データを上記メモリに上記記憶データとして記憶することを特徴とする請求項1に記載の半導体装置。

【請求項3】 上記メモリは、上記複数の演算部に関連して設けられた書き込みおよび読み出し用の第1のポートと、読み出し専用の第2のポートとを有することを特徴とする請求項1に記載の半導体装置。

【請求項4】 上記メモリをクリアするためのデータを生成し、該データによって上記メモリをクリアする手段を備えることを特徴とする請求項1に記載の半導体装置。

【請求項5】 上記メモリをプリセットするためのデータを生成し、該データによって上記メモリをプリセットする手段を備えることを特徴とする請求項1に記載の半導体装置。

【請求項6】 上記複数の演算部による演算結果がオーバフローとなる場合、上記複数の演算部に対応した上記メモリの所定領域に最大値をセットする手段を備えることを特徴とする請求項1に記載の半導体装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、例えば動きベクトル検出装置における差分絶対値の足し込み演算の処理部に適用して好適な半導体装置に関する。詳しくは、演算器およびメモリが一体化され、演算器を構成する複数の演算部はそれぞれメモリのカラムのピッチに揃えて配されることによって、演算器とメモリとの間のデータのやり取りを効率よく行うことができるようにした半導体装置に係るものである。

【0002】

【従来の技術】画像処理においては、動きベクトル検出は重要な要素の一つであり、その代表的な方法としてブロックマッチング法がある。これは、あるフレームの一部を構成するある画素ブロック（参照ブロック）について、時間の異なるフレームにおける様々な位置での同一形状画素ブロック（候補ブロック）との相関を評価し、その中で相関が最も高い候補ブロックとの間の相対的な位置ずれを、その参照ブロックにおける動きベクトルとみなすものである。

【0003】ここで、候補ブロックを想定する領域が探索範囲である。相関の評価には、参照ブロックと候補ブロックとの対応する各画素間の画素データの差分絶対値

のブロック内各画素についての総和、すなわち差分絶対値和が用いられることが多い。1個の参照ブロックにつき探索範囲内の画素数分の差分絶対値和（相関値）が得られるが、これが相関値テーブルである。この相関値テーブルの中で最も差分絶対値和の小さい、すなわち相関の高いところが、画素を単位とした動きベクトルと見なされる。また実際には、この処理は演算負荷の非常に重いものであり、ブロックの形状や大きさ、あるいは演算に使用する画素位置などについて、様々な工夫が行われている。

【0004】

【発明が解決しようとする課題】従来のブロックマッチングには、差分絶対値演算器と記憶素子としての複数のレジスタとを組み合わせた、あるいはさらに加算器をも組み合わせたPE (Processing Element)が多く用いられている。アレイ配置したPE間に並列かつパイプライン的にデータを流すことで、複数の差分絶対値和を並列に演算し、または差分絶対値を得た後に加算器によって総和をし、差分絶対値和（相関値）の集まり、すなわち相関値テーブルを生成するのである。

【0005】この場合、記憶素子としてレジスタを用いるためにその構成素子数が多く占有面積が広くなり、さらに各PEに複数個（例えば2～3個）のレジスタがあるためPE全体としても占有面積が広くなる。そのため、半導体チップが大型化するという問題点があった。

【0006】この問題点を解決するために、記憶素子として、レジスタよりも小型、高密度な半導体メモリを用いることが考えられる。その場合、加算器と半導体メモリとの間でデータのやり取りをして、差分絶対値の足し込み演算を行って、最終的に相関値（差分絶対値和）を得ることになる。

【0007】そこで、この発明では、複数の演算部とメモリとの間のデータのやり取りを効率よく行い得る半導体装置を提供することを目的とする。

【0008】

【課題を解決するための手段】この発明に係る半導体装置は、演算器およびメモリが一体化されてなり、演算器を構成するビット単位の複数の演算部はそれぞれメモリのカラムのピッチに揃えて配されているものである。

【0009】この発明においては、演算器とメモリとが一体化されている。そして、演算器を構成するビット単位の複数の演算部が、それぞれメモリのカラムのピッチに揃えて配されている。例えば、演算器は加算器であり、入力データとメモリの記憶データとを複数の演算部で加算し、この加算データをメモリに記憶データとして記憶するものである。

【0010】このように、演算器およびメモリが一体化され、演算器を構成するビット単位の複数の演算部がメモリのカラムのピッチに揃えて配されるものであり、演

算器からメモリへの演算データの供給および半導体メモリから演算器への記憶データの供給が効率的に行われる。

【0011】なお、メモリが、上述の複数の演算部に関連して設けられた書き込みおよび読み出し用の第1のポートと読み出し専用の第2のポートとを有することで、メモリの記憶データの読み出しを、演算器による演算とは別途独立して行うことができる。

【0012】また、メモリをクリアまたはプリセットするためのデータを生成し、このデータによってメモリをクリアまたはプリセットすることで、外部からクリアまたはプリセットするためのデータを入力することなく、メモリを容易にクリアまたはプリセットすることができる。

【0013】また、演算器を構成する複数の演算部による演算結果がオーバフローとなる場合、複数の演算部に対応したメモリの所定領域に最大値をセットすることで、この半導体メモリの所定領域に演算結果として誤った小さな値が記憶されることを防止できる。

【0014】

【発明の実施の形態】以下、図面を参照しながら、この発明の実施の形態について説明する。図1は、実施の形態としての動き補償予測符号化装置100の構成を示している。この符号化装置100は、画像データ（動画画像を構成するフレームデータ）Diを入力する入力端子101と、この入力端子101に供給される画像データDiと後述する動き補償回路110から供給される予測画像データとの差分を演算する減算器102と、この減算器102で得られる差分データに対してDCT（離散コサイン変換）を行うDCT回路103と、このDCT回路103で得られるDCT係数に対して量子化を行う量子化回路104と、この量子化回路104で得られた符号化データDoを出力する出力端子105とを有している。

【0015】また、符号化装置100は、量子化回路104で得られた符号化データDoに対して逆量子化を行う逆量子化回路106と、この逆量子化回路106の出力データに対して逆DCTを行って差分データを得る逆DCT回路107と、この逆DCT回路107で得られる差分データと動き補償回路110で得られる予測画像データとを加算して元の画像データを復元する加算器108と、この加算器108で復元された画像データを記憶するフレームメモリ109とを有している。

【0016】また、符号化装置100は、フレームメモリ109に記憶された画像データを読み込み、後述する動きベクトル検出回路111からの動きベクトルMVに基づいて動き補償をした後、上述したように減算器102および加算器108に予測画像データとして供給する動き補償回路110と、入力端子101に供給される画像データDiの動きベクトルMVを検出して動き補償回

路110に供給する動きベクトル検出回路111とを有している。

【0017】図1に示す動き補償予測符号化装置100の動作を説明する。入力端子101に入力される画像データDiは、減算器102および動きベクトル検出回路111に供給される。減算器102では、この画像データDiと動き補償回路110から供給される予測画像データとの差分が演算される。

【0018】減算器102で得られる差分データはDCT回路103に供給されて離散コサイン変換される。このDCT回路103で得られるDCT係数は量子化回路104に供給されて量子化される。そして、この量子化回路104で得られた符号化データDoが出力端子105に出力される。

【0019】また、量子化回路104で得られた符号化データDoが逆量子化回路106に供給されて逆量子化され、さらにこの逆量子化回路106の出力データが逆DCT回路107に供給されて逆DCTされ、差分データが復元される。この差分データと動き補償回路110からの予測データとを加算器108で加算されて元の画像データが復元され、この復元された画像データがフレームメモリ109に記憶される。

【0020】動き補償回路110では、あるフレームにおいては、その前のフレームにフレームメモリ109に記憶された画像データの読み込みが行われて、動きベクトル検出回路111からの動きベクトルMVに基づいて動き補償されて、予測画像データが得られる。この予測画像データは、上述したように、差分データを得るために減算器102に供給されると共に、画像データを復元するために加算器108に供給される。

【0021】次に、動きベクトル検出回路111の詳細を説明する。この動きベクトル検出回路111では、ブロックマッチング法により動きベクトルが検出される。これは、図2に示すように、探索フレームの候補ブロックを所定の探索範囲内で移動し、参照フレームの参照ブロックと最も合致している候補ブロックを検出することにより、動きベクトルを求めるものである。

【0022】ブロックマッチング法では、図3Aに示すように、1枚の画像、例えば水平H画素、垂直Vラインの1フレームの画像が図4Bに示すように、P画素×Qラインのブロックに細分化される。図3Bの例では、P=5、Q=5の例である。cがブロックの中心画素位置である。

【0023】図4A～Cは、cを中心画素とする参照ブロックとc'を中心とする候補ブロックの位置関係を示している。cを中心画素とする参照ブロックは、参照フレームの注目しているある参照ブロックであり、それと一致する探索フレームの候補ブロックが探索フレームにおいてc'を中心とするブロックの位置にあるものとしている。ブロックマッチング法では、探索範囲内におい

て、参照ブロックと最も合致する候補ブロックを見出すことによって、動きベクトルを検出する。

【0024】図4Aの場合では、水平方向に+1画素、垂直方向に+1ライン、すなわち、(+1, +1)の動きベクトルが検出される。図4Bでは、(+3, +3)の動きベクトルMVが検出され、図4Cでは、(+2, -1)の動きベクトルが検出される。動きベクトルは、参照フレームの参照ブロック毎に求められる。

【0025】動きベクトルを探索する範囲を水平方向で±S画素、垂直方向で±Tラインとすると、参照ブロックは、その中心c'に対して、水平に±S、垂直に±Tずれたところに中心c''を有する候補ブロックと比較される必要がある。

【0026】図5は、参照フレームのある参照ブロックの中心c'の位置をRとする時に、比較すべき探索フレームの(2S+1)×(2T+1)個の候補ブロックとの比較が必要なことを示している。すなわち、この図5のます目の位置にc''が存在する候補ブロックの全てが比較対象である。図5は、S=4、T=3とした例である。

【0027】探索範囲内の比較で得られた評価値(すなわち、フレーム差の絶対値和、このフレーム差の二乗和、あるいはフレーム差の絶対値のn乗和等)の中で、最小値を検出することによって、動きベクトルが検出される。図5の探索範囲は、候補ブロックの中心が位置する領域であり、候補ブロックの全体が含まれる探索範囲の大きさは、(2S+P)×(2T+Q)となる。

【0028】図6は、動きベクトル検出回路111の構成を示している。この動きベクトル検出回路111は、回路全体の動作を制御するシステムコントローラ121と、参照フレームの画像データDiが入力される入力端子122と、この画像データDiを探索フレームの画像データとして蓄積する探索フレームメモリ123とを有している。フレームメモリ123の書き込み、読み出し等の動作は、システムコントローラ121によって制御される。

【0029】また、動きベクトル検出回路111は、複数の差分絶対値演算器124₁~124_Nを有している。ここで、Nは、ある参照ブロック内の1個の画素データに対する探索範囲に存在する複数の画素データの個数である。換言すれば、Nは、ある参照ブロックに対する探索範囲に存在する複数の候補ブロックの個数である。複数の演算器124₁~124_Nは、入力端子122に入力される画像データDiを構成する画素データを、参照ブロックの画素データDrとして共通に入力すると共に、また当該画素データDrに対する探索範囲に存在する、複数の候補ブロックの画素データDc₁~Dc_Nをそれぞれ入力し、参照ブロックの画素データと候補ブロックの画素データとの差分絶対値を演算するものである。

【0030】この場合、演算器124₁~124_Nにおいては、図7に示すように、1個の参照画素とN個の探索範囲画素との1対Nのマッチング演算が行われる。ここで、参照ブロック内における参照画素の位置に応じて、この参照画素に対する探索範囲画素の位置が変化する。例えば、ハッチングして示した位置は、参照ブロックの左上の1個の画素に対するN個の探索範囲画素の位置を示している。

【0031】また、動きベクトル検出回路111は、複数の加算器125₁~125_Nと、複数の記憶領域126₁~126_Nを有する相関値テーブル生成用の半導体メモリ126とを有している。複数の加算器125₁~125_Nは、複数の演算器124₁~124_Nで演算されて得られた差分絶対値をそれぞれ入力すると共に、半導体メモリ126の複数の記憶領域126₁~126_Nに記憶されていた記憶データのそれぞれを入力し、差分絶対値を記憶データに足し込むものである。

【0032】このように、複数の加算器125₁~125_Nで得られた足し込みデータのそれぞれは、半導体メモリ126の複数の記憶領域126₁~126_Nに記憶データとして書き戻される。半導体メモリ126の書き込み、読み出しの動作は、システムコントローラ121によって制御される。

【0033】システムコントローラ121は、参照フレームの各参照ブロックについて、複数の差分絶対値演算器124₁~124_Nにおける差分絶対値の演算、複数の加算器125₁~125_Nにおける足し込みの演算、半導体メモリ126の複数の記憶領域126₁~126_Nへの足し込みデータの書き戻しを、ブロック内画素分だけ行い、半導体メモリ126の複数の記憶領域126₁~126_Nに、各参照ブロックに対する探索範囲に存在する複数の候補ブロックのそれぞれに対応した相関値が得られるように制御する。

【0034】なお、入力端子122に入力される画像データDiは各ラインの画素データが連続したものとなっている。そのため、演算器124₁~124_Nに入力される参照ブロックの画素データDrは、参照ブロック毎に連続したものではなく、複数の参照ブロックの画素データが所定数ずつ連続したものとなっている。例えば、参照ブロックが、図3Bに示すようにP画素×Qラインで構成される場合には、あるラインの画素データは、P画素毎に異なった参照ブロックを構成している。また、ある参照ブロックに着目すると、当該参照ブロックの画素データは、Qラインの画素データが入力されてはじめて、全て入力されることとなる。

【0035】このように、演算器124₁~124_Nに入力される参照ブロックの画素データDrは複数の参照ブロックの画素データが所定数ずつ連続したものとなっていることから、上述した複数の差分絶対値演算器124₁~124_Nにおける差分絶対値の演算、複数の加算

器125₁~125_nにおける足し込みの演算、半導体メモリ126の複数の記憶領域126₁~126_nへの足し込みデータの書き戻しは、複数の参照ブロックに対応して、時分割的に行われる。そして、Qラインの画素データの入力毎に、新たな複数の参照ブロックの処理に移っていく。

【0036】また、動きベクトル検出回路111は、参照ブロック毎に、半導体メモリ126の複数の記憶領域126₁~126_nに得られた、参照ブロックに対する探索範囲に存在する複数の候補ブロックのそれぞれに 10 対応した相関値（差分絶対値和）に基づいて、参照ブロックに対応した動きベクトルMVを検出する相関値テーブル評価器127と、この評価器127で検出された動きベクトルMVを出力する出力端子128とを有している。評価器127では、最小の相関値を発生する候補ブロックの位置を、動きベクトルMVとして検出する。

【0037】図6に示す動きベクトル検出回路111の動作を説明する。入力端子122に入力される画像データDiは、参照ブロックの画素データDrとして、複数の差分絶対値演算器124₁~124_nに共通に入力さ 20 される。また、入力端子122に入力される画像データDiはフレームメモリ123に供給され、探索フレームの画像データとして蓄積される。

【0038】また、フレームメモリ123から当該参照ブロックに対する探索範囲に存在する複数の候補ブロックの画素データDc₁~Dc_nが複数の差分絶対値演算器124₁~124_nにそれぞれ入力される。この候補 30 ブロックの画素データDc₁~Dc_nは、それぞれ参照ブロックの画素データDrに対応した画素位置のものとされる。演算器124₁~124_nでは、画素データDrと画素データDc₁~Dc_nとの差分絶対値がそれぞれ演算される。

【0039】また、複数の演算器124₁~124_nで演算されて得られた差分絶対値はそれぞれ複数の加算器125₁~125_nに入力される。また、この複数の加算器125₁~125_nには、半導体メモリ126の複数の記憶領域126₁~126_nに記憶されていた記憶データがそれぞれ入力される。後述するように、複数の記憶領域126₁~126_nのそれぞれは複数の参照ブ 40 ロック分の記憶部からなっている。上述したように複数の加算器125₁~125_nに入力される記憶データは、画素データDrが含まれる参照ブロックに対応した記憶部より読み出される。

【0040】複数の加算器125₁~125_nでは、それぞれ記憶データに差分絶対値が足し込まれる。そして、このように複数の加算器125₁~125_nで得られた足し込みデータのそれぞれは、半導体メモリ126の複数の記憶領域126₁~126_nに記憶データとして書き戻される。この場合、画素データDrが含まれる参照ブロックに対応した記憶部 50 に書き戻される。

【0041】上述した複数の差分絶対値演算器124₁~124_nにおける差分絶対値の演算、複数の加算器125₁~125_nにおける足し込みの演算、半導体メモリ126の複数の記憶領域126₁~126_nへの足し込みデータの書き戻しは、参照フレームの各参照ブロックについて、ブロック内画素分だけ行われる。これにより、半導体メモリ126の複数の記憶領域126₁~126_nに、各参照ブロックについて、参照ブロックに対する探索範囲に存在する複数の候補ブロックのそれぞれ 10 に対応した相関値（差分絶対値和）が得られる。相関値テーブル評価器127は、半導体メモリ126の複数の記憶領域126₁~126_nに得られた、各参照ブロックに対する探索範囲に存在する複数の候補ブロックのそれぞれに対応した相関値は、順次読み出されて相関値テーブル評価器127に供給される。評価器127では、各参照ブロックについて、最小の相関値を発生する候補ブロックの位置が、動きベクトルMVとして検出される。このように、評価器127で検出された各参照ブ 20 ロックにおける動きベクトルMVは順次出力端子128に出力される。

【0042】なお、本実施の形態において、複数の加算器125₁~125_nと半導体メモリ126は一体化されており、複数の加算器125₁~125_nを構成するそれぞれのビット単位の複数の加算部は、半導体メモリ126のカラムのピッチに揃えて配されている。

【0043】図8は、加算器125₁およびそれに対応した半導体メモリ126の記憶領域126₁の部分の詳細構成を示したものである。なお、説明は省略するが、加算器125₁~125_nおよびそれに対応した半導体メモリ126の記憶領域126₁~126_nの部分につ 30 いても同様に構成されている。

【0044】図8において、記憶領域126₁には、カラム方向にn個、ロウ方向にX+1個のメモリセル(Memory Cell)130がマトリックス状に配されている。この場合、カラム方向に延びる各行のn個のメモリセル130によって、それぞれ1参照ブロック分の記憶部が構成されている。

【0045】図9は、メモリセル130の構成例を示している。このメモリセル130は、書き込みおよび読み出し用の第1のポートと読み出し専用の第2のポートとを有する2ポート構成のものである。

【0046】負荷素子であるP型MOSトランジスタQ1とN型MOSトランジスタQ3とが電源と接地との間に直列に接続されてCMOSインバータ11が形成されていると共に、負荷素子であるP型MOSトランジスタQ2とN型MOSトランジスタQ4とが電源と接地との間に直列に接続されてCMOSインバータ12が形成されている。そして、これらCMOSインバータ11、12の各出力、すなわち記憶ノードN1、N2の各電位が 50 互いに他のCMOSインバータ12、11の入力、すな

わちN型MOSトランジスタQ4、Q3の各ゲート入力となっている。

【0047】CMOSインバータ11の記憶ノードN1は、ゲートが端子13に接続されたアクセストランジスタQ5を介して端子14に接続される。一方、CMOSインバータ12の記憶ノードN2は、ゲートが端子13に接続されたアクセストランジスタQ6を介して端子15に接続される。端子13にはワード線WLが接続され、端子14にはビット線BLが接続され、端子15にはビット線/BL (/BLはBLバーを表している)が接続される。

【0048】また、N型MOSトランジスタQ7、Q8が直列に接続され、その一端は接地され、その他端は端子16に接続される。そして、トランジスタQ7のゲートは記憶ノードN1に接続され、トランジスタQ8のゲートは端子17に接続される。端子16には読み出し専用ビット線BRLが接続され、端子17には読み出し専用ワード線WRLが接続される。

【0049】このようなメモリセル130において、一対のCMOSインバータ11、12で構成されるメモリセル部に“1”または“0”のデータが記憶される。そして、このメモリセル部とビット線BL、/BLとの間で、アクセストランジスタQ5、Q6を介して、読み出しおよび書き込みのデータ転送が行われる。また、メモリセル部と読み出し専用ビット線BRLとの間で、アクセストランジスタQ8を介して読み出しのデータ転送が行われる。

【0050】なお、図9に示すメモリセル130の構成例は、SRAM(Static Random Access Memory)セルをベースとしたものであるが、他のメモリセル、例えばDRAM(Dynamic Random Access Memory)、FeRAM(Ferroelectric Random Access Memory)、MRAM(Magnetic Random Access Memory)等におけるメモリセルをベースにして構成してもよい。

【0051】図8に戻って、カラム方向に並ぶ各行のメモリセル130に沿って、ワード線WL₀~WL_x、および読み出し専用ワード線WRL₀~WRL_xが配されている。上述したように、ワード線WL₀~WL_xはメモリセル130の端子13に接続され、読み出し専用ワード線WRL₀~WRL_xはメモリセル130の端子17に接続される。

【0052】また、ロウ方向に並ぶ各列のメモリセル130に沿って、ビット線BL₀~BL_{n-1}、/BL₀~/BL_{n-1}、および読み出し専用ビット線BRL₀~BRL_{n-1}が配されている。上述したように、ビット線BL₀~BL_{n-1}はメモリセル130の端子14に接続され、ビット線/BL₀~/BL_{n-1}はメモリセル130の端子15に接続され、読み出し専用ビット線BRL₀~BRL_{n-1}はメモリセル130の端子16に接続される。

【0053】なお、この読み出し専用ビット線BRL₀、

~BRL_{n-1}による読み出しモードに入る前には、ビット線BRL₀~BRL_{n-1}をプリチャージすることが必要となる。そのために、ビット線BRL₀はP型MOSトランジスタQ41を介して電源に接続される。そして、このトランジスタQ41のゲートには、プリチャージ制御信号/ ϕ_{rpc} (ϕ_{rpc} は ϕ_{rpc} バーを表しており、プリチャージ制御信号 ϕ_{rpc} が反転されたものである)が入力される。ビット線BRL₁~BRL_{n-1}に関しても同様に構成されている。

【0054】また、メモリ領域126₁のロウ方向に並ぶ各列のメモリセル130にそれぞれ対応して、センスアンプSA₀~SA_{n-1}が配されている。各センスアンプSA₀~SA_{n-1}は、それぞれビット線BL₀~BL_{n-1}、/BL₀~/BL_{n-1}に接続されている。これにより、記憶領域126₁のロウ方向に並ぶ各列のメモリセル130から、ビット線対BL₀、/BL₀~BL_{n-1}、/BL_{n-1}およびセンスアンプSA₀~SA_{n-1}を介して記憶データMD₀~MD_{n-1}の読み出しが行われる。

【0055】ここで、センスアンプSA₀の部分の構成の詳細を説明する。ビット線BL₀は、P型MOSトランジスタQ21を介してN型MOSトランジスタQ22のゲートに接続される。また、ビット線/BL₀は、P型MOSトランジスタQ23を介してN型MOSトランジスタQ24のゲートに接続される。そして、トランジスタQ22、Q24の互いのソースは接続され、その接続点はN型MOSトランジスタQ25を介して接地される。そして、トランジスタQ21、Q23のゲートには、読み出し制御信号/ ϕ_{r} (ϕ_{r} は ϕ_{r} バーを表しており、読み出し制御信号 ϕ_{r} が反転されたものである)が入力され、トランジスタQ25のゲートには、イコライズ制御信号/ ϕ_{eq} (ϕ_{eq} は ϕ_{eq} バーを表しており、イコライズ制御信号 ϕ_{eq} が反転されたものである)が入力される。

【0056】また、トランジスタQ22のドレインはP型MOSトランジスタQ26、Q27の並列回路を介して電源に接続され、トランジスタQ24のドレインはP型MOSトランジスタQ28、Q29の並列回路を介して電源に接続される。そして、トランジスタQ22のドレインはトランジスタQ29のゲートに接続され、トランジスタQ24のドレインはトランジスタQ27のゲートに接続される。トランジスタQ26、Q28のゲートには、イコライズ制御信号/ ϕ_{eq} が入力される。

【0057】なお、読み出しモードに入る前にはビット線対BL₀、/BL₀をプリチャージすることが必要となる。そのために、ビット線BL₀はP型MOSトランジスタQ31を介して電源に接続され、ビット線/BL₀はP型MOSトランジスタQ32を介して電源に接続され、ビット線BL₁、/BL₁はP型MOSトランジスタQ33を介して接続される。そして、トランジスタQ31~Q33のゲートには、プリチャージ制御信号/ ϕ_{pc}

が入力される。

【0058】センスアンプ $SA_1 \sim SA_{n-1}$ の部分の構成も、上述したセンスアンプ SA_0 の部分の構成と同様とされる。

【0059】また、上述したように、カラム方向に延びる各行の n 個のメモリセル130によって、それぞれ1つの参照ブロックの記憶部が構成されている。所定の記憶部にある参照ブロックの足し込みデータを順次書き込むことを開始する前に、この所定の記憶部を構成するメモリセル130の記憶データをクリアすることが必要となる。そのために、ビット線対 $BL_0, /BL_0 \sim BL_{n-1}, /BL_{n-1}$ のそれぞれに対応して、“0”のデータを生成し、このデータをメモリセル130に書き込みデータとして供給する構成を備えている。

【0060】すなわち、ビット線 BL_0 はN型MOSトランジスタQ51を介して接地される。そして、このトランジスタQ51のゲートには、クリア制御信号 ϕ_{clr} が入力される。ビット線対 $BL_1, /BL_1 \sim BL_{n-1}, /BL_{n-1}$ の部分に関しても同様に構成されている。

【0061】また、加算器125₁は n ビットのそれぞれのビットの加算を行うための n 個の加算部140₀～140_{n-1}からなっており、これら n 個の加算部140₀～140_{n-1}はメモリ領域126₁のカラムのピッチに揃えて配されている。

【0062】加算部140₀～140_{n-1}のそれぞれのA側の入力端子には、差分絶対値演算器124₁からの8ビットの差分絶対値のビットデータ $D_0 \sim D_7$ が入力される。また、加算部140₀～140_{n-1}のそれぞれのA側の入力端子は接地され、“0”が入力された状態とされる。一方、加算部140₀～140_{n-1}のそれぞれのB側の入力端子には、これら加算部140₀～140_{n-1}のそれぞれに対応して記憶領域126₁のロウ方向に並ぶメモリセル130から、ビット線対 $BL_0, /BL_0 \sim BL_{n-1}, /BL_{n-1}$ およびセンスアンプ $SA_0 \sim SA_{n-1}$ を介して読み出された記憶データ $MD_0 \sim MD_{n-1}$ がそれぞれ入力される。

【0063】加算部140₀の非反転出力端子 S は、N型MOSトランジスタQ11のゲートに接続されている。そして、このトランジスタQ11のドレインは、加算部140₀に対応してロウ方向に並ぶメモリセル130に接続されているビット線/ BL_0 に接続される。一方、この加算部140₀の反転出力端子/ S （/ S は S バーを表している）は、N型MOSトランジスタQ12のゲートに接続される。そして、このトランジスタQ12のドレインは、加算部140₀に対応してロウ方向に並ぶメモリセル130に接続されているビット線 BL_0 に接続される。

【0064】トランジスタQ11、Q12の互いのソースは接続され、その接続点はN型MOSトランジスタQ13、Q14の直列回路を介して接地される。そして、

トランジスタQ14のゲートには書き込み制御信号 ϕ_w が入力され、トランジスタQ13のゲートには加算部140₀のキャリ出力端子 C_{out} に得られるMSB (Most Significant Bit)のキャリ出力 C_{msb} がインバータ141を介して入力される。

【0065】加算部140₁～140_{n-1}の出力端子 S 、/ S 側の構成も、上述した加算部140₀の出力端子 S 、/ S 側の構成と同様とされる。

【0066】また、加算部140₀のキャリ入力端子 C_{in} は接地され、“0”が入力された状態とされる。また、加算部140₀～140_{n-1}のキャリ出力端子 C_{out} は、それぞれ加算部140₁～140_{n-1}に接続されている。これにより、加算部140₀～140_{n-1}で n ビット加算器が構成される。

【0067】また、ビット線/ BL_0 はN型MOSトランジスタQ61、Q62を介して接地される。そして、トランジスタQ61のゲートにはクリア制御信号/ ϕ_{clr} （/ ϕ_{clr} は ϕ_{clr} バーを表しており、クリア制御信号 ϕ_{clr} が反転されたものである）が入力され、トランジスタQ62のゲートには加算部140_{n-1}のキャリ出力端子 C_{out} に得られるMSBのキャリ出力 C_{msb} が入力される。

【0068】図8に示す加算器125₁および記憶領域126₁の部分の動作を説明する。まず、カラム方向に延びる各行の n 個のメモリセル130によってそれぞれ1つの参照ブロックの記憶部が構成されているが、所定の記憶部を構成するメモリセル130の記憶データをクリアする動作について説明する。

【0069】所定の記憶部を構成するメモリセル130の記憶データをクリアする場合、書き込み制御信号 ϕ_w およびクリア制御信号 ϕ_{clr} はアクティブ、つまり“1”とされ、読み出し制御信号 ϕ_r およびイコライズ制御信号 ϕ_{eq} はインアクティブ、つまり“0”とされ、さらにワード線 $WL_0 \sim WL_x$ のうち、所定の記憶部に対応するワード線が活性化される。

【0070】この場合、クリア制御信号 ϕ_{clr} がアクティブとされてトランジスタQ51がオンとなる。そのため、“0”のデータが生成され、このデータがビット線 $BL_0 \sim BL_{n-1}$ に出力される。したがって、所定の記憶部に対応するワード線を活性化することで、当該所定の記憶部を構成する n 個のメモリセル130には“0”のデータが書き込まれ、記憶データのクリアが行われる。

【0071】次に、所定の記憶部に記憶されている記憶データ $MD_0 \sim MD_{n-1}$ に、8ビットの差分絶対値 $D_0 \sim D_7$ を、加算器125₁（加算部140₁～140_{n-1}）で足し込み、そして加算器125₁で得られた足し込みデータ $AD_0 \sim AD_{n-1}$ を、当該所定の記憶部に書き戻す動作について説明する。

【0072】所定の記憶部に記憶されている記憶データ $MD_0 \sim MD_{n-1}$ に、8ビットの差分絶対値 $D_0 \sim D_7$ を足

10

20

30

40

50

し込む場合、最初に、イコライズ制御信号 ϕ_{eq} はアクティブ、つまり“1”とされ、書き込み制御信号 ϕ_w 、読み出し制御信号 ϕ_r およびクリア制御信号 ϕ_{clr} はインアクティブ、つまり“0”とされ、ビット線対 BL_0 、/ $BL_0 \sim BL_{n-1}$ 、/ BL_{n-1} のイコライズ（ブリチャージ）が行われる。

【0073】この場合、ビット線対 BL_0 、/ BL_0 に関しては、イコライズ制御信号 ϕ_{eq} がアクティブとされてトランジスタQ31～Q33の全てがオンとなり、ビット線 BL_0 およびビット線/ BL_0 に電源の電位が印加され、これらビット線 BL_0 およびビット線/ BL_0 は同電位となる。他のビット線対 BL_1 、/ $BL_1 \sim BL_{n-1}$ 、/ BL_{n-1} に関しても同様である。

【0074】このようにビット線対 BL_0 、/ $BL_0 \sim BL_{n-1}$ 、/ BL_{n-1} のイコライズが行われた状態で、読み出し制御信号 ϕ_r はアクティブ、つまり“1”とされ、書き込み制御信号 ϕ_w 、イコライズ制御信号 ϕ_{eq} およびクリア制御信号 ϕ_{clr} はインアクティブ、つまり“0”とされ、さらにワード線 $WL_0 \sim WL_x$ のうち、所定の記憶部に対応するワード線が活性化される。

【0075】これにより、所定の記憶部を構成するn個のメモリセル130の記憶データ $MD_0 \sim MD_{n-1}$ が、それぞれビット線対 BL_0 、/ $BL_0 \sim BL_{n-1}$ 、/ BL_{n-1} およびセンスアンプ $SA_0 \sim SA_{n-1}$ を介して読み出され、加算部140₀～140_{n-1}のB側の入力端子にそれぞれ入力される。したがって、所定の記憶部に記憶されている記憶データ $MD_0 \sim MD_{n-1}$ に、8ビットの差分絶対値 $D_0 \sim D_x$ が足し込まれる。

【0076】そして、加算部140₀～140_{n-1}における加算出力、つまり足し込みデータ $AD_0 \sim AD_{n-1}$ が有効になったところで、書き込み制御信号 ϕ_w はアクティブ、つまり“1”とされ、読み出し制御信号 ϕ_r 、イコライズ制御信号 ϕ_{eq} およびクリア制御信号 ϕ_{clr} はインアクティブ、つまり“0”とされ、さらにワード線 $WL_0 \sim WL_x$ のうち、所定の記憶部に対応するワード線が活性化される。

【0077】この場合、加算部140₀の部分に関して、足し込みデータ S_0 が“1”である場合には、トランジスタQ11はオン、トランジスタQ12はオフとなり、ビット線/ BL_0 に“0”が出力されることから、所定の記憶部を構成するn個のメモリセル130のうち、当該加算部140₀に対応するメモリセル130には、“1”のデータが記憶される。一方、加算部140₀の部分に関して、足し込みデータ S_0 が“0”である場合には、トランジスタQ11はオフ、トランジスタQ12はオンとなり、ビット線 BL_0 に“0”が出力されることから、所定の記憶部を構成するn個のメモリセル130のうち、当該加算部140₀に対応するメモリセル130には、“0”のデータが記憶される。

【0078】他の加算部140₁～140_{n-1}の部分に関

しても同様である。これにより、加算器125₁で得られた足し込みデータ $AD_0 \sim AD_{n-1}$ は、所定の記憶部を構成するn個のメモリセル130に書き戻される。

【0079】なお、足し込みの動作において、オーバーフローとなる場合には、加算部140_{n-1}のキャリ出力端子 C_{out} に得られるMSBのキャリ出力 C_{msb} が“1”となるため、トランジスタQ13はオフとなり、足し込みデータ $AD_0 \sim AD_{n-1}$ が、所定の記憶部を構成するn個のメモリセル130に書き込まれることはない。

【0080】その代わり、この場合、トランジスタQ61がオンとなる他に、トランジスタQ62もオンとなるので、ビット線/ $BL_0 \sim /BL_{n-1}$ にそれぞれ“0”の信号が出力される。したがって、所定の記憶部を構成するn個のメモリセル130のそれぞれに“1”のデータが書き込まれる。つまり、この所定の記憶部には最大値が記憶される。

【0081】次に、所定の記憶部に記憶された、ある参照ブロックに対応した最終的な足し込みデータ、つまり相関値（差分絶対値和）を、読み出す場合の動作を説明する。

【0082】最初に、ブリチャージ制御信号 ϕ_{brc} がアクティブ、つまり“1”とされ、読み出し専用ビット線 $BR_0 \sim BR_{n-1}$ のブリチャージが行われる。この場合、トランジスタQ41はオンとなり、読み出し専用ビット線 $BR_0 \sim BR_{n-1}$ のそれぞれに電源の電位が印加される。

【0083】このように読み出し専用ビット線 $BR_0 \sim BR_{n-1}$ のイコライズが行われた状態で、読み出し専用ワード線 $WR_0 \sim WR_x$ のうち、所定の記憶部に対応する読み出し専用ワード線が活性化される。これにより、所定の記憶部を構成するn個のメモリセル130の記憶データ $\Sigma_0 \sim \Sigma_{n-1}$ が、それぞれ読み出し専用ビット線 $BR_0 \sim BR_{n-1}$ に得られる。ここで、記憶データ $\Sigma_0 \sim \Sigma_{n-1}$ は、nビットの相関値（差分絶対値和）を構成している。

【0084】以上説明したように、本実施の形態においては、加算器125₁～125_nと相関値テーブル生成用の半導体メモリ126とを用いて差分絶対値を累積していき、この半導体メモリ126に、参照ブロックに対する探索範囲に存在する複数の候補ブロックのそれぞれに対応した相関値（差分絶対値和）を得るようにしたものであり、従来のように記憶素子としてレジスタを用いるものに比べて、占有面積を小さくでき、半導体チップの大型化を防止することができる。

【0085】また、加算器125₁～125_nおよび相関値テーブル生成用の半導体メモリ126とを一体化し、加算器125₁～125_nをそれぞれ構成するビット単位の複数の加算部140₀～140_{n-1}が半導体メモリ126のカラムのピッチに揃えて配されているので（図8参照）、加算器125₁～125_nから半導体メ

メモリ126への足し込みデータ $AD_0 \sim AD_{n-1}$ の供給および半導体メモリ126から加算器125 $_1 \sim 125_n$ への記憶データ $MD_0 \sim MD_{n-1}$ の供給を効率的に行うことができる。

【0086】また、半導体メモリ126が、上述の複数の加算部140 $_0 \sim 140_{n-1}$ に関連して設けられた書き込みおよび読み出し用の第1のポートと、読み出し専用の第2のポートとを有するものであり（図8参照）、半導体メモリ126からのある参照ブロックに対応する相関値 $\Sigma_0 \sim \Sigma_{n-1}$ の読み出しを、足し込みとは別途独立して行うことができる。

【0087】また、半導体メモリ126の所定の記憶部を構成するメモリセル130の記憶データをクリアする際に、トランジスタQ51をオンとして、クリアするための“0”データを生成し、このデータをメモリセル130に書き込みデータとして供給するものであり、外部からクリアするためのデータを入力することなく、半導体メモリ126を容易にクリアすることができる。

【0088】また、加算器125 $_1 \sim 125_n$ をそれぞれ構成する複数の加算部140 $_0 \sim 140_{n-1}$ による演算結果がオーバフローとなる場合、複数の加算部140 $_0 \sim 140_{n-1}$ に対応した半導体メモリ126の所定の記憶部に最大値を記憶（セット）するものであり、この所定の記憶部に誤った小さな値が相関値として記憶され、動きベクトルの検出誤りが生じることを防止できる。

【0089】なお、上述実施の形態において、半導体メモリ126の記憶領域126 $_1 \sim 126_n$ では、ビット線 $BL_0 \sim BL_{n-1}$ をトランジスタQ51を介して接地し、このトランジスタQ51のゲートにクリア制御信号/ ϕ_{clr} を入力する構成とし、クリア信号 ϕ_{clr} をアクティブとしたとき所定の記憶部を構成するn個のメモリセル130に“0”のデータが書き込まれてクリアされるものを示した（図8参照）。

【0090】ここで、図8に破線図示するように、ビット線/ $BL_0 \sim /BL_{n-1}$ をトランジスタQ52を介して接地し、このトランジスタQ52のゲートにクリア制御信号/ ϕ_{clr} を入力する構成とした場合、クリア信号 ϕ_{clr} をアクティブとしたとき、トランジスタQ52で“1”のデータが生成され、所定の記憶部を構成するn個のメモリセル130に“1”のデータが書き込まれる。

【0091】したがって、ビット線対 $BL_0, /BL_0 \sim BL_{n-1}, /BL_{n-1}$ のそれぞれに対応してトランジスタQ51、Q52を設けておき、それらのいずれかを選択的にビット線に接続し、クリア信号 ϕ_{clr} をアクティブとしたとき所定の記憶部を構成するn個のメモリセル130に、所定のデータをプリセットするようにしてもよい。このプリセットデータを工夫することで、例えば平坦な絵柄の部分においては（0，0）等の特定の動きベクトルが検出されやすくなる。このプリセットの設定は、半導体装置の設計時に予め決めておけばよく、従っ

てコンタクトレイヤのプログラムなどによって設定することが想定される。

【0092】また、上述実施の形態においては、加算器125 $_1 \sim 125_n$ および相関値テーブル生成用の半導体メモリ126とを一体化したものであるが、さらに差分絶対値演算器124 $_1 \sim 124_n$ や相関値テーブル評価器127をも一体化するようにしてもよい。

【0093】また、上述実施の形態においては、メモリセル130が2ポート構成のものであったが（図9参照）、メモリセルが2ポート構成でなくてもよく、半導体メモリ126全体として2ポート構成であってもよい。さらには、半導体メモリ126として2ポート構成でなくとも、例えば映像信号におけるブランキング期間に相関値（テーブルデータ）の読み出しを行ったり、同一機能ブロックを複数個有してフィールドまたはフレーム間でインターリーブさせて用いるなど、足し込みと相関値の読み出しとを同一ポートにおいて異なる期間に行ってもよい。

【0094】また、上述実施の形態においては、加算器125 $_1 \sim 125_n$ および半導体メモリ126を用いた足し込みを、動きベクトル検出における差分絶対値の足し込みに適用したものであるが、他の信号処理における同様の足し込みにも適用することができる。

【0095】また、上述実施の形態においては、加算器125 $_1 \sim 125_n$ および相関値テーブル生成用の半導体メモリ126とを一体化したものであるが、減算器、乗算器、除算器などのその他の演算器と半導体メモリとを一体化したものも同様に構成でき、演算器と半導体メモリとの間のデータのやり取りを効率よく行うことができる。

【0096】次に、図10以降を使用して、動きベクトル検出回路の他の例について説明する。最初に、この例における動きベクトル検出方法について説明する。従来のブロックマッチング法は、参照ブロックと各候補ブロックとの間の対応する画素データ間の差分絶対値をブロックについて全て加算した差分絶対値和を相関値として得、これによって生成される探索範囲に対応した大きさの相関値テーブルにおいて、テーブルの中心位置と相関値が最小である位置とのずれをその参照ブロックにおける動きベクトルとするものである。

【0097】この方法においては、マッチングのとれていない情報をも全て含めて相関値テーブルを生成しており、相関値の分布は広がったものとなる。そのため、図10に示すように、参照ブロック内に動きの異なるものが複数混在する場合には、各々の動きを示す相関値の最小値が他の動きにおける分布に埋もれ、誤った位置に最小値が生じることもある。

【0098】ここで、従来のブロックマッチング法の処理手順を説明する。図11の中央列に示すように、参照ブロックを構成する画素データと、これに対応する探索

範囲内の各画素データとの間で1対Nのマッチング演算、例えば差分絶対値演算を行って、暫定的な相関値テーブルを得る。そして、参照ブロックを構成する#1～#Mの全画素データに対応した暫定相関値テーブルの各差分絶対値を、探索範囲の画素位置毎に足し込んで、相関値テーブルを生成する。そして、この相関値テーブルに基づいて参照ブロックの動きベクトルを検出する。

【0099】これに対して、この例における係る動きベクトル検出方法では、図11の右列に示すように、参照ブロックを構成する各画素データ毎に、マッチング演算の結果に基づき、マッチングの度合いがその閾値より高いときは計数値として0以外の値を配し、マッチングの度合いがその閾値以下のときは計数値として0を配した計数値テーブルを生成する。

*

表1 計数値の生成例

差分絶対値	計数値	補正した計数値
0	3	$k \times 3$
1	2	$k \times 2$
2	1	k
3以上	0	0

【0102】そして、参照ブロックを構成する#1～#Mの全画素データに対応した計数値テーブルの各計数値を度数とし、この度数を探索範囲の画素位置毎に足し込んで、探索範囲の画素位置に対応して度数の累積値が配されたヒストグラムテーブルを生成する。そして、このヒストグラムテーブルの度数の極大値または極小値に基づいて、参照ブロックの動きベクトルを検出する。

【0103】ここで、マッチングの度合いが高いときに配する計数値が正の値であるときは極大値であり、一方その計数値が負の値であるときは極小値である。このヒストグラムテーブルにおける度数の極大値または極小値の位置は、参照ブロック内に存在する動きに対応した動きベクトルを表すものとなる。

【0104】この例の動きベクトル検出方法によれば、マッチングの度合いが高いときのみ計数値として0以外の値を配して計数値テーブルを生成するものであって、ヒストグラムテーブルはマッチング度合いが高いもののみによって生成されることとなり、その度数の分布はより先鋭化されたものとなる。そのため、参照ブロック内に動きの異なるものが複数混在する場合には、ヒストグラムテーブルに複数の極大値または極小値が明瞭に分離して示され易くなる。したがって、参照ブロックの一個または複数の動きベクトルを正しく検出できる。

【0105】参照ブロック内に異なる2つの動きがある場合について、従来のブロックマッチング法における相

*【0100】図示の計数値テーブルにおいては、最もマッチングの度合いが高い画素位置に○を記し、次にマッチングの度合いが高い画素値に△を記し、またその次にマッチングの度合いが高い画素位置に×を記している。そして、計数値としては、○が「+3」、△が「+2」、×が「+1」、印のない画素位置は「0」などとする。例えば、表1に示すように、マッチング演算結果としての例えば差分絶対値が0、1、2のときは、それぞれ計数値として「+3」、「+2」、「+1」を生成し、一方その差分絶対値が3以上であるときは計数値として「0」を生成する。

【0101】

【表1】

関値テーブルおよびこの例の動きベクトル検出方法におけるヒストグラムテーブルの一例を、それぞれ図12および図13に示す。図12に示す相関値テーブルは、この発明に係るヒストグラムテーブルとの比較を容易にするため、相関値の大小関係を入れ換えて表示している。すなわち、いずれにおいても、最大値あるいは極大値の位置が動きベクトルを表す。

【0106】図12に示す相関値テーブルに基づいて動きベクトルを検出する従来のブロックマッチング法では、より支配的な方の動きを示す動きベクトルは検出されるが、他方の動きを示す動きは埋もれてしまって、その動きを示す動きベクトルの検出は困難である。これに対して、図13に示すヒストグラムテーブルに基づいて動きベクトルを検出するこの例の動きベクトル検出方法では、ヒストグラムテーブルに明瞭に分離して2個の極大値が現れており、2つの動きをそれぞれ示す2個の動きベクトルを容易に検出できる。

【0107】なお、上述では、マッチングの度合いを上位3位とそれ以外との4種類に分類したが、これに限る必要はなく、3種類以下あるいは5種類以上であっても構わない。また、上述では、各計数値をマッチングの度合いに応じて小さい方から「0」、「+1」、「+2」、「+3」としたが、これについてもこの計数値の用い方に限るものではなく、例えば○～×については全て「+1」とし、それ以外を「0」としてもよい。ただ

し、マッチングの度合いがその閾値より高いときに配する計数値を一定値とするのではなく、マッチングの度合いが高い程、絶対値の大きな値とすることで、ヒストグラムテーブルの度数の分布を一層先鋭化できる利益がある。

【0108】ところで、参照ブロックを構成する画素データの周囲におけるアクティビティが高い場合、すなわちその画素データとその周囲にある画素データとの値の違いが大きい場合（信号波形としてみた場合に波形振幅が大きい場合）、本来正しいとすべき動きベクトルの位置においてもマッチングの度合いが低くなってしまふ。これは、一般的には1画素よりも小さな動き量の成分があり、そのため周囲画素データとの値の違いが大きいほど、マッチングすべき位置においても画素データ値間の差分が大きくなってしまふことによる。

【0109】この場合の対処方法として、周囲におけるアクティビティが大きな参照ブロックに対応した計数値テーブルの計数値に関しては、その計数値の絶対値が大

きくなるように設定することが有効である。

【0110】具体的には、まず、参照ブロックを構成する各画素データ毎に、アクティビティAをそれぞれ求める。例えば、図14に示すように、アクティビティAを求める対象の画素データをaとし、その周囲の4個の画素データをb～dとし、(1)式により、アクティビティAを算出する。

$$A = |a-b| + |a-c| + |a-d| + |a-e| \quad \dots (1)$$

【0111】そして、参照ブロックを構成する各画素データ毎に、アクティビティAの最大アクティビティA_{max}に対する割合が高い程大きな値を補正係数kとして得るようにする。表2は、補正係数kの設定例を示している。この表2において、A_{nor}は、アクティビティAを最大アクティビティA_{max}で割って正規化したアクティビティであって、アクティビティAの最大アクティビティA_{max}に対する割合を示すものである。

【0112】

【表2】

表2 補正係数kの設定例

$A_{nor} = A / A_{max}$	補正係数 k
$0.75 < A_{nor}$	6
$0.5 < A_{nor} \leq 0.75$	4
$0.25 < A_{nor} \leq 0.5$	2
$A_{nor} \leq 0.25$	1

【0113】そして、参照ブロックを構成する各画素データに対応した計数値テーブルの計数値に、それぞれこの各画素データに対応した補正係数kを掛けて補正する（表1の「補正した計数値」の項参照）。

【0114】なお、上述したように、補正係数kを求め、この補正係数kを計数値に掛けて補正する代わりに、マッチングの度合いに応じて計数値を生成する際に用いられる閾値を、アクティビティが大きい程低くなるように変更してもよい。

【0115】例えば、上述の表1の例では、差分絶対値が閾値3より小さいときは計数値を0以外の値（+3～+1）とし、一方差分絶対値が閾値3以上のときは計数値を0としたが、この閾値をアクティビティが大きい程大きくするように変更する。この場合、差分絶対値の閾値を大きくするということは、マッチングの度合いの閾値を低くすることを意味している。

【0116】このように、閾値を変更する場合であっても、上述した計数値を補正係数kで補正する場合と同様に、周囲におけるアクティビティが大きな参照ブロックの画素データに対応した計数値テーブルの計数値に関しては、その絶対値が大きくなる。これにより、アクティ

ビティによるマッチングの度合いの変化による不具合を軽減できる。

【0117】上述したように、この例に係る動きベクトル検出方法では、ヒストグラムテーブルに基づいて、参照ブロックの1個または複数個の動きベクトルを検出できる。なお、このように検出された動きベクトルに基づいて、さらに参照ブロックを構成する各画素データの動きベクトルを特定することができる。

【0118】その場合、参照ブロックを構成する画素データと、この画素データの画素位置を基準として上述したように検出された動きベクトルに対応した探索フレームの画素位置の画素データとの間の相関情報、例えば差分絶対値あるいは差分二乗値等を求め、最も相関が高くなる場合の動きベクトルを、その参照ブロックを構成する画素データの動きベクトルとする。

【0119】上述したように、参照ブロックの動きベクトルを正しく検出できることから、その動きベクトルに基づいて特定される各画素データの動きベクトルも精度の高いものとなる。なお、このように特定される参照ブロックを構成する各画素データの動きベクトルから孤立した動きベクトルを除去することで、この各画素データ

の動きベクトルの精度をさらに高めることができる。

【0120】ここで、孤立した動きベクトルとは、ある画素データの動きベクトルがその周囲の画素データの動きベクトルと異なっている場合のそのある画素データの動きベクトルを意味している。そして、孤立した動きベクトルの除去とは、そのある画素データの動きベクトルをその周囲の画素データの動きベクトルに置き換えることを意味している。

【0121】図15は、上述した他の例としての動きベクトル検出回路200の構成を示している。この動きベクトル検出回路200においては、図16に示すように、探索フレームに対して参照フレームは過去のフレームであり、入力フレームが探索フレームである。

【0122】この動きベクトル検出回路200は、回路全体の動作を制御するシステムコントローラ201と、探索フレームとしての入力フレームの画像データDiが入力される入力端子202と、この画像データDiを、次フレームの参照フレームの画像データとして蓄積する参照フレームメモリ203とを有している。フレームメモリ203の書き込み、読み出し等の動作は、システム

コントローラ201によって制御される。

【0123】また、動きベクトル検出回路200は、マッチング演算としての差分絶対値演算を行う差分絶対値演算器204₁～204_Nを有している。複数の演算器204₁～204_Nは、入力端子202に入力される画像データDiを構成する画素データを探索フレームの画素データDcとして共通に入力すると共に、この画素データDcを探索範囲に含む参照フレームの所定の参照ブロック（同一タイミングでは最大2個）の画素データDr₁～Dr_Nをそれぞれ入力し、画素データDcと画素データDr₁～Dr_Nとの差分絶対値を演算するものである。

【0124】この場合、図16に示すように、探索フレーム内の入力画素とマッチング演算をするべき参照フレーム内の参照画素は、その探索範囲内に入力画素が位置する複数の参照画素である。その複数の参照画素の位置する範囲は、ある参照画素の探索範囲と表裏一体の関係にあり、入力画素を仮に参照画素と見なした場合の探索範囲を左右および上下で反転したものとなる。

【0125】従って、これらの複数の参照画素は、多くの場合は複数の参照ブロックにわたって位置し、また1個の参照ブロックにおける画素数も1個から全てと一定ではない。また、これら参照ブロックの組み合わせおよび1個の参照ブロックにおける画素の組み合わせは、入力画素の位置によっても異なる。

【0126】また、図15に戻って、動きベクトル検出回路200は、複数の演算器204₁～204_Nで演算されて得られた複数の差分絶対値のそれぞれに対応して、計数値を生成する計数値生成部205を有している。この計数値生成部205は、計数値として、差分絶

対値がその閾値より小さいときは、マッチングの度合いがその閾値より高いとして0以外の所定値を生成し、一方差分絶対値がその閾値以上のときは、マッチングの度合いがその閾値以下であるとして0を生成する。例えば、計数値生成部205は、上述した表1に示すように、差分絶対値が0、1、2のときは、それぞれ計数値として「+3」、「+2」、「+1」を生成し、一方その差分絶対値が3以上であるときは計数値として「0」を生成する。この場合、閾値は3である。

【0127】また、動きベクトル検出回路200は、複数の演算器204₁～204_Nにそれぞれ対応した記憶領域206₁～206_Nを有する、計数値テーブル生成用の記憶手段としての半導体メモリ206を有している。半導体メモリ206の書き込み、読み出しの動作は、システムコントローラ201によって制御される。

【0128】システムコントローラ201は、複数の演算器204₁～204_Nで得られた差分絶対値にそれぞれ対応して計数値生成部205で生成された計数値を、その計数値に係る演算器および参照ブロックの画素位置に対応した半導体メモリ206の計数値記憶領域に書き込み、この半導体メモリ206に、参照ブロックを構成する各画素データに対応した計数値テーブルが得られるように制御する。ここで、計数値テーブルは、探索範囲の各画素位置に対応して計数値が配されたものである。

【0129】図17に示すように、この動きベクトル検出回路200において、参照ブロックの大きさは8×8画素、探索範囲は(−3～+4)×(−3～+4)画素とされる。参照ブロックの探索範囲には、(3+4+1)×(3+4+1)=64個の候補ブロックが存在する。そのため、N=64とされる。つまり、複数の演算器204₁～204_N、半導体メモリ206の複数の記憶領域206₁～206_Nは、それぞれ64個である。

【0130】複数の演算器204₁～204_Nは、各参照ブロックについて、それぞれ1個の候補ブロックに対応した差分絶対値を求める演算を担当する。また、半導体メモリ206の複数の記憶領域206₁～206_Nは、それぞれ参照フレームの横一行の参照ブロック数に対応した個数のブロック記憶領域を有している。さらに、各ブロック記憶領域は、参照ブロックを構成する画素データの個数分の計数値記憶領域を有している。

【0131】なおここで、入力端子202に入力される画像データDiはプログレッシブ方式によるものとする。入力端子202に入力される画像データDiは各ラインの画素データが連続したものとなっている。

【0132】図17に示すように、探索範囲は(−3～+4)×(−3～+4)画素であり、また参照ブロックの大きさは8×8画素である。そのため、ある参照ブロックに対する探索範囲の最も上の入力画素が含まれるラインを1ライン目とすると、この探索範囲の最も下の入力画素が含まれるラインは15ライン目となる。したが

って、ある参照ブロックを構成する各画素データに対応した計数値テーブルを得るためには、15ラインの画素データが必要となる。なお、図17には、1個の参照ブロックのみを示しているが、上述の15ラインの画素データに基づいて、横一行分の全参照ブロックについて、それを構成する各画素データに対応した計数値テーブルを得ることができる。

【0133】また、動きベクトル検出回路200は、参照ブロックを構成する各画素データ毎に周囲におけるアクティビティAを求め、この各画素データ毎にアクティビティAの最大アクティビティA_{max}に対する割合が高い程大きな値を補正係数kとして得る補正係数取得部207を有している。

【0134】補正係数取得部207は、例えば以下のようにして、参照ブロックを構成する各画素データにそれぞれ対応した補正係数kを取得する。まず、例えば図14に示すように、アクティビティAを求める対象の画素データをaとし、その周囲の4個の画素データをb〜dとし、上述した(1)式により、参照ブロックを構成する各画素データ毎に、アクティビティAを求める。次に、参照ブロックを構成する各画素データ毎に、アクティビティAを最大アクティビティA_{max}で割って正規化したアクティビティA_{nor}を求め、例えば上述した表2のように補正係数kを設定する。

【0135】また、動きベクトル検出回路200は、上述した半導体メモリ206より読み出され、後述する複数の加算器209₁〜209_nに供給される、参照ブロックを構成する各画素データに対応した計数値テーブルの計数値に、それぞれ補正係数取得部207で取得された各画素データに対応した補正係数kを掛けて補正する計数値補正部208を有している。

【0136】また、動きベクトル検出回路200は、複数の加算器209₁〜209_nと、複数の記憶領域210₁〜210_nを有するヒストグラムテーブル生成用の記憶手段としての半導体メモリ210とを有している。複数の加算器209₁〜209_nは、半導体メモリ206より読み出され、計数値補正部208で補正された、参照ブロックを構成する所定の画素データに対応した計数値テーブルの計数値をそれぞれ度数として入力すると共に、半導体メモリ210の複数の記憶領域210₁〜210_nに記憶されていた記憶データのそれぞれを入力し、度数を記憶データに足し込むものである。

【0137】このように、複数の加算器209₁〜209_nで得られた足し込みデータのそれぞれは、半導体メモリ210の複数の記憶領域210₁〜210_nに記憶データとして書き戻される。半導体メモリ210の書き込み、読み出しの動作は、システムコントローラ201によって制御される。

【0138】システムコントローラ201は、半導体メモリ206からの計数値テーブルの計数値の読み出し、

複数の加算器209₁〜209_nにおける足し込みの演算、半導体メモリ210の複数の記憶領域210₁〜210_nへの足し込みデータの書き戻しを、参照ブロックを構成する各画素データに対応した複数の計数値テーブルの個数分だけ繰り返し、半導体メモリ210に、探索範囲の各画素位置に対応して度数の累積値が配されたヒストグラムテーブルが得られるように制御する。

【0139】また、動きベクトル検出回路200は、参照ブロック毎に、半導体メモリ210に得られたヒストグラムテーブルにおける極大値に基づいて、動きベクトルを検出する動きベクトル検出部211を有している。ヒストグラムテーブルにおける度数の極大値は、参照ブロック内に存在する動きに対応した動きベクトルを表している。そのため、参照ブロック内に動きの異なるものが複数混在する場合にはヒストグラムテーブルに複数の極大値が示されることとなり、動きベクトル検出部211では複数の動きベクトルが検出される。

【0140】また、動きベクトル検出回路200は、動きベクトル検出部211で検出された、参照ブロックの一個または複数の動きベクトルMV₁〜MV_n。(nは1以上の整数)に基づいて、その参照ブロックを構成する各画素データの動きベクトルmvを特定する動きベクトル特定部212と、この動きベクトル特定部212で特定された各画素データの動きベクトルmvを出力する出力端子213とを有している。

【0141】動きベクトル特定部212は、参照ブロックを構成する画素データとこの画素データの画素位置を基準として、動きベクトル検出部211で検出された各動きベクトルMV₁〜MV_nに対応した探索フレームの画素位置の画素データとの間の相関情報、例えば差分絶対値あるいは差分二乗値を求め、最も相関が高くなる場合の動きベクトルをこの画素データの動きベクトルmvとする。

【0142】図32は、動きベクトル特定部212の具体的な構成を示している。この動きベクトル特定部212は、全体の動作を制御するためのコントローラ221を有している。このコントローラ221には、入力端子202(図15参照)に入力される入力フレーム(探索フレーム)の画像データDiが供給され、この画像データDiは探索画素用のメモリ222に蓄積される。また、このコントローラ221には、動きベクトル検出部211(図15参照)で検出される動きベクトルMV₁〜MV_nが供給される。

【0143】また、動きベクトル特定部212は、相関演算部223₁〜223_nを有している。この相関演算部223₁〜223_nには、参照ブロックを構成する画素データ(参照画素データ)が、コントローラ221の制御によって、参照フレームメモリ203から読み出されて共通に供給される。また、この相関演算部223₁〜223_nには、それぞれ、その参照ブロックを構成す

る画素データの画素位置を基準として動きベクトル $MV_1 \sim MV_n$ に対応した探索フレームの画素位置の画素データ(探索画素データ)が、コントローラ221の制御によって、メモリ222から読み出されて供給される。相関演算部223₁~223_nは、それぞれ、参照画素データと探索画素データとの間の相関情報、例えば差分絶対値あるいは差分二乗値等を求める。

【0144】また、動きベクトル特定部212は、動きベクトル仮特定部224を有している。この仮特定部224には、動きベクトル検出部211で検出される動きベクトル $MV_1 \sim MV_n$ が供給される。また、この仮特定部224には、相関演算部223₁~223_nで求められた相関情報が供給される。この仮特定部224は、相関演算部223₁~223_nで求められた相関情報に基づいて、最も相関が高くなる場合の動きベクトルを、上述の参照画素データの動きベクトル m_v として特定する。

【0145】なお、この場合、どの動きベクトル $MV_1 \sim MV_n$ についても相関が充分でない場合には、動きベクトル m_v を出力する代わりに、動きが不定である旨の情報を出力する。この動きが不定である参照画素データとしては、ノイズが重畳した画素データあるいは動きのない部分の画素データが考えられる。

【0146】また、動きベクトル特定部212は、仮特定部224で特定された参照ブロックを構成する各画素データの動きベクトル m_v から孤立した動きベクトルを除去する等の補正処理を行う動きベクトル補正部225を有している。ここで、孤立した動きベクトルとは、上述したように、ある画素データの動きベクトルがその周囲の画素データの動きベクトルと異なっている場合のそのある画素データの動きベクトルを意味している。そして、孤立した動きベクトルの除去とは、そのある画素データの動きベクトルをその周囲の画素データの動きベクトルに置き換えることを意味している。

【0147】図18に示す動きベクトル特定部212の動作を説明する。動きベクトル検出部211で検出された、所定の参照ブロックの動きベクトル $MV_1 \sim MV_n$ が、コントローラ221および動きベクトル仮特定部224に供給される。

【0148】相関演算部223₁~223_nには、上述した所定の参照ブロックを構成する画素データ(参照画素データ)が参照フレームメモリ203から読み出されて共通に供給される。また、この相関演算部223₁~223_nには、それぞれその参照ブロックを構成する画素データの画素位置を基準として動きベクトル $MV_1 \sim MV_n$ に対応した探索フレームの画素位置の画素データ(探索画素データ)がメモリ222から読み出されて供給される。そして、相関演算部223₁~223_nでは、それぞれ、参照画素データと探索画素データとの間の相関情報が求められる。

【0149】相関演算部223₁~223_nで求められた相関情報は仮特定部224に供給される。この仮特定部224では、相関情報に基づいて、最も相関が高くなる場合の動きベクトルが、参照画素データの動きベクトル m_v として特定される。この場合、どの動きベクトル $MV_1 \sim MV_n$ についても相関が充分でない場合には、動きベクトル m_v の代わりに、動きが不定である旨の情報が出力される。

【0150】以上の動作が、参照ブロックを構成する各画素データの全てに対して行われることで、この各画素データの動きベクトル m_v が特定される。このように仮特定部224で特定された参照ブロックを構成する各画素データの動きベクトル MV (動き不定情報も含む)は、動きベクトル補正部225で孤立した動きベクトルの除去(孤立点除去)等の補正処理が行われた後、順次あるいは同時に出力される。

【0151】図18に示す動きベクトル特定部212では、動きベクトル m_v を1画素データずつ得るものであるが、相関演算部223₁~223_nおよび動きベクトル仮特定部224の組を複数組備えて、動きベクトル m_v を複数画素データ分並行して得るようにしてもよい。これにより、例えば、参照ブロックを構成する全ての画素データ分を並行して得ることができる。

【0152】図15に示す動きベクトル検出回路200の動作を説明する。入力端子202に入力される画像データ D_i を構成する画素データは、探索フレーム(入力フレーム)の画素データ D_c として、複数の差分絶対値演算器204₁~204_nに共通に入力される。また、この画像データ D_i は、フレームメモリ203に供給され、次フレームで使用する参照フレームの画像データとして蓄積される。

【0153】また、複数の演算器204₁~204_nには、フレームメモリ203から、画素データ D_c を探索範囲に含む、参照フレームの所定の参照ブロックの画素データ $D_{r_1} \sim D_{r_n}$ がそれぞれ入力される。そして、複数の演算器204₁~204_nでは、画素データ D_c と画素データ $D_{r_1} \sim D_{r_n}$ との差分絶対値がそれぞれ演算される。この場合、複数の演算器204₁~204_nでは、各参照ブロックについて、それぞれ1個の候補ブロックに対応した差分絶対値を求める演算が行われる。

【0154】複数の演算器204₁~204_nで得られた差分絶対値はそれぞれ計数値生成部205に供給される。計数値生成部205では、差分絶対値のそれぞれに対応して、差分絶対値がその閾値より小さいときは0以外の所定値を生成し、一方差分絶対値がその閾値以上のときは0を生成する(表1参照)。

【0155】複数の演算器204₁~204_nで得られた差分絶対値に対応して計数値生成部205で生成された計数値は、それぞれ計数値テーブル生成用の半導体メ

メモリ206の記憶領域206₋₁~206_{-n}に書き込みデータとして供給される。

【0156】上述したように、各記憶領域206₋₁~206_{-n}には、それぞれ参照フレームの横一行の参照ブロック数に対応した個数のブロック記憶領域を有している。さらに、各ブロック記憶領域には、参照ブロックを構成する画素データの個数分の計数値記憶領域を有している。

【0157】複数の演算器204₋₁~204_{-n}で得られた差分絶対値にそれぞれ対応して計数値生成部205で生成された計数値は、その計数値に係る演算器および参照ブロックの画素位置に対応した、半導体メモリ206の記憶領域に書き込まれる。これにより、半導体メモリ206の各ブロック領域に、参照ブロックを構成する各画素データに対応した、探索範囲の各画素位置に対応して計数値が配された計数値テーブルが得られる。

【0158】補正係数取得部207では、参照ブロックを構成する各画素データ毎に周囲におけるアクティビティAが求められ、そしてこの各画素データ毎にアクティビティAの最大アクティビティA_{max}に対する割合が高い程大きな値が補正係数kとして取得される(図14、表2参照)。

【0159】半導体メモリ206に得られる参照ブロックを構成する各画素データに対応した計数値テーブルの計数値(N個)は順次読み出され、計数値補正部208を介して、複数の加算器209₋₁~209_{-n}に供給される。計数値補正部208では、参照ブロックを構成する各画素データに対応した計数値テーブルの計数値に、それぞれ補正係数取得部207で取得された各画素データに対応した補正係数kが掛算されて補正が行われる。

【0160】複数の加算器209₋₁~209_{-n}には、半導体メモリ206より読み出され、計数値補正部208で補正された、参照ブロックを構成する所定の画素データに対応した計数値テーブルの計数値がそれぞれ度数として入力され、また半導体メモリ210の複数の記憶領域210₋₁~210_{-n}に記憶されていた記憶データのそれぞれが入力される。

【0161】そして、複数の加算器209₋₁~209_{-n}では、度数が記憶データに足し込まれる。複数の加算器209₋₁~209_{-n}で得られた足し込みデータのそれぞれは、半導体メモリ210の複数の記憶領域210₋₁~210_{-n}に記憶データとして書き戻される。

【0162】半導体メモリ206からの計数値テーブルの計数値の読み出し、複数の加算器209₋₁~209_{-n}における足し込みの演算、半導体メモリ210の複数の記憶領域210₋₁~210_{-n}への足し込みデータの書き戻しは、参照ブロックを構成する各画素データに対応した複数の計数値テーブルの個数分だけ繰り返し行われる。これにより、半導体メモリ210に、探索範囲の各画素位置に対応して度数の累積値が配された、当該参照

ブロックに対応したヒストグラムテーブルが得られる。

【0163】動きベクトル検出部211では、半導体メモリ210に得られる各参照ブロックに対応したヒストグラムにおける極大値に基づいて、各参照ブロックの動きベクトルMV₁~MV_nが検出される。上述したように、ヒストグラムテーブルにおける度数の極大値は、参照ブロック内に存在する動きに対応した動きベクトルを表している。そのため、参照ブロック内に動きの異なるものが複数混在する場合にはヒストグラムテーブルに複数の極大値が示され、従って動きベクトル検出部211では複数の動きベクトルが検出される。

【0164】このように動きベクトル検出部211で検出される、参照ブロックの動きベクトルMV₁~MV_nは動きベクトル特定部212に供給される。この動きベクトル特定部212では、動きベクトルMV₁~MV_nに基づいて、その参照ブロックを構成する各画素データの動きベクトルmvが特定される。そして、この各画素データの動きベクトルmvは出力端子213に出力される。

【0165】なお、上述したように、15ラインの画素データに基づいて、横一行分の全参照ブロックについて、それを構成する各画素データに対応した計数値テーブルを得ることができる。この場合、この15ラインの画素データの後半側のラインの画素データは参照フレームの次の横一行分の参照ブロックについての演算処理にも使用される。そのため、その後半側のラインの画素データの入力時には、次の横一行分の参照ブロックについての演算処理も並行して行う必要がある。したがって、実際には、図15に示す動きベクトル検出回路200における、差分絶対値演算器204₋₁~204_{-n}、計数値生成部205、半導体メモリ206、計数値補正部208、加算器209₋₁~209_{-n}、半導体メモリ210の部分は、少なくとも2系統設けられることとなる。

【0166】次に、差分絶対値演算器204₋₁~204_{-n}に関してさらに説明する。図19Aは、入力画素と演算器204₋₁~204_{-n}((0,0)~(7,7))の演算対称範囲の配置を示している。図19Bは、参照ブロックを構成する8×8個の画素データ(a, a)~(h, h)を示している。

【0167】入力画素と差分絶対値演算をすべき参照ブロックの画素データの位置する範囲(演算対象範囲)は、各演算器(0,0)~(7,7)が担当する範囲に分けられる。各演算器(0,0)~(7,7)の担当範囲は1画素である。

【0168】図20に示すように、入力画素が参照ブロックの探索範囲の1ライン目にあり、その演算対象範囲が参照ブロックの画素データ(a, a)にかかるとき、当該参照ブロックを構成する8×8個の画素データに対応する計数値テーブルを得るための演算が始まる。このとき、演算器(7,7)にはフレームメモリ203(図15に図示)から画素データ(a, a)が入力される。

10

20

30

40

50

そして、この演算器(7, 7)で、入力画素と画素データ(a, a)との差分絶対値が求められる。

【0169】次に、図21に示すように、1画素期間後には、入力画素が1画素だけ右のものに移り、入力画素の演算対象範囲が参照ブロックの画素データ(b, a)にかかる。このとき、演算器(7, 7)にはフレームメモリ203から画素データ(b, a)が入力される。そして、この演算器(7, 7)で、入力画素と画素データ(b, a)と差分絶対値が求められる。このとき並行して、演算器(6, 7)にはフレームメモリ203から画素データ(a, a)が入力される。そして、この演算器(6, 7)で、入力画素と画素データ(a, a)との差分絶対値が求められる。

【0170】以下、1ライン目については、入力画素が右隣のものに順次移っていき、担当範囲に参照ブロックの画素データを含む演算器では、それぞれ入力画素と参照ブロックの画素データとの演算が並行して行われる。

【0171】図22は、7画素期間後の状態を示している。この状態では、入力画素が7画素だけ右のものに移り、入力画素の演算対象範囲が参照ブロックの画素データ(h, a)までかかる。このとき、演算器(7, 7)～(0, 7)には、それぞれフレームメモリ203から画素データ(h, a)～(a, a)が入力される。そして、この演算器(7, 7)～(0, 7)で、それぞれ入力画素と画素データ(h, a)～(a, a)との差分絶対値が求められる。

【0172】また、図23に示すように、入力画素が参照ブロックに対する探索範囲の2ライン目にあり、その演算対象範囲が参照ブロックの画素データ(a, a), (a, b)にかかるとき、演算器(7, 7)にはフレームメモリ203から画素データ(a, b)が入力される。そして、この演算器(7, 7)で、入力画素と画素データ(a, b)との差分絶対値が求められる。このとき並行して、演算器(7, 6)にはフレームメモリ203から画素データ(a, a)が入力される。そして、この演算器(7, 6)で、入力画素と画素データ(a, a)との差分絶対値が求められる。

【0173】以下、2ライン目については、入力画素が右隣のものに順次移っていき、担当範囲に参照ブロックの画素データを含む演算器では、それぞれ入力画素と参照ブロックの画素データとの演算が並行して行われる。

【0174】以下の3ライン目～15ライン目についても同様であり、入力画素が右隣のものに順次移っていき、担当範囲に参照ブロックの画素データを含む演算器では、それぞれ入力画素と参照ブロックの画素データとの演算が並行して行われる。

【0175】図24は、入力画素が参照ブロックに対する探索範囲の8ライン目にあり、その演算対象範囲が参照ブロックの画素データ(a, a)～(a, h)までの8個の画素データにかかった状態を示している。この状

態では、演算器(7, 7)～(7, 0)には、フレームメモリ203から画素データ(a, h)～(a, a)が入力される。そして、この演算器(7, 7)～(7, 0)で、入力画素と画素データ(a, h)～(a, a)との差分絶対値が求められる。

【0176】図25は、入力画素が参照ブロックに対する探索範囲の15ライン目にあり、その演算対象範囲が参照ブロックの左下の画素データ(a, h)にかかっている状態を示している。この状態では、演算器(7, 0)には、フレームメモリ203から画素データ(a, h)が入力される。そして、この演算器(7, 0)で、入力画素と画素データ(a, h)との差分絶対値が求められる。

【0177】図26は、上述した各演算器204₁～204_n((0, 0)～(7, 7))における各ラインの差分絶対値演算の過程を示している。この図26において、1～64の数字はそれぞれ入力画素と参照ブロックの画素データ(a, a)～(h, h)との演算を示している。

【0178】各演算器(0, 0)～(7, 7)では、ある参照ブロックに関連して、それぞれ、入力画素データと参照ブロックを構成する64個の画素データ(a, a)～(h, h)との差分絶対値演算が行われる。つまり、各演算器(0, 0)～(7, 7)では、それぞれ、ある参照ブロックとその参照ブロックに対する探索範囲に存在する1個の候補ブロックとの間の、対応する画素データ同士による64個の差分絶対値演算が行われる。図27は、一例として、演算器(7, 7)における演算過程を示している。

【0179】図28は、動きベクトル(1個の参照画素に対する探索範囲の座標)を示している。参照ブロックの探索範囲には、動きベクトル(x, y)のそれぞれに対応した64個の候補ブロックが存在する。そして、参照ブロックを構成する各画素データに対応した計数値テーブルは、それぞれ、動きベクトル(x, y)のそれぞれに対応した64個の計数値からなっている。

【0180】図29は、半導体メモリ206のある1個の参照ブロックに対応した領域における、その参照ブロックを構成する各画素データ(a, a)～(h, h)に対応したそれぞれ計数値テーブルにおける64個の計数値(それぞれ動きベクトル(+4, +4)～(-3, -3)に対応している)が、どの計数値記憶領域(アドレス)に記録されるかを示している。

【0181】またこの図29は、各演算器(0, 0)～(7, 7)のそれぞれにおける演算によって、どの動きベクトルに対応した計数値を生成するための差分絶対値が得られるかを示している。すなわち、演算器(0, 0)では、(+4, +4)の動きベクトルに対応した計数値を生成するための差分絶対値が求められる。同様に、演算器(1, 0)～(7, 7)では、それぞれ(+3, +4)

〜(−3, −3)の動きベクトルに対応した計数値を生成するための差分絶対値が求められる。

【0182】なお、ある参照ブロックに対応して各演算器204_{−1}〜204_{−N}((0, 0)〜(7, 7))では上述したように演算が行われるが、ある参照ブロックに関する演算と並行して、このある参照ブロック(参照ブロックP)に対して水平方向の前後に隣接する参照ブロックに関する演算も行われる。すなわち、図26から明らかなように、参照ブロックPに関する演算とその前後の参照ブロックP−1, P+1に関する演算とが、時間的にだぶって行われていることがわかる。

【0183】例えば、参照ブロックPに関する演算は各ラインの15画素を用いて行われる。1〜7番目の画素を用いるときは、参照ブロックPに関する演算が行われると共に、これと並行して参照ブロックP−1に関する演算が行われる。同様に、9〜15番目の画素を用いるときは、参照ブロックPに関する演算が行われると共に、これと並行して参照ブロックP+1に関する演算が行われる。なお、8番目の画素を用いるときは、参照ブロックPに関する演算のみが行われる。

【0184】例えば、1ライン目で、1番目の画素を用いるとき、演算器(7, 7)では参照ブロックPに関する演算が行われるが、演算器(6, 7)〜(0, 7)では参照ブロックP−1に関する演算が行われる。また例えば、1ライン目で、9番目の画素を用いるときは、演算器(6, 7)〜(0, 7)では参照ブロックPに関する演算が行われるが、演算器(7, 7)では参照ブロックP+1に関する演算が行われる。

【0185】上述したように半導体メモリ206の複数の記憶領域206_{−1}〜206_{−N}には、それぞれ参照フレームの横一行の参照ブロック数に対応した個数のブロック記憶領域を有している。さらに、各ブロック記憶領域には、参照ブロックを構成する画素データの個数分の計数値記憶領域を有している。そして、複数の演算器204_{−1}〜204_{−N}で得られた差分絶対値にそれぞれ対応して計数値生成部205で生成された計数値は、その計数値に係る演算器および参照ブロックの画素位置に対応した半導体メモリ206の計数値記憶領域に書き込まれる。このことから、半導体メモリ206は、カラム方向には、複数の演算器204_{−1}〜204_{−N}のそれぞれに対応した所定メモリセル単位で、独立してアクセス可能に構成される。

【0186】次に、半導体メモリ206の詳細構成について説明する。図30は、半導体メモリ206の全体構成を示している。半導体メモリ206は、8個のメモリ部250_{−1}〜250_{−8}から構成されている。各メモリ部250_{−1}〜250_{−8}は、それぞれ読み出し用のロウデコーダ251と、8個の記憶領域と、書き込み用のメインロウデコーダ253と、書き込み用のサブロウデコーダ254とから構成されている。

【0187】メモリ部250_{−1}, 250_{−2}, 250_{−3}, 250_{−4}, 250_{−5}, 250_{−6}, 250_{−7}, 250_{−8}の8個の記憶領域は、それぞれ、演算器204_{−1}〜204_{−8}, 204_{−9}〜204_{−16}, 204_{−17}〜204_{−24}, 204_{−25}〜204_{−32}, 204_{−33}〜204_{−40}, 204_{−41}〜204_{−48}, 204_{−49}〜204_{−56}, 204_{−57}〜204_{−64}にそれぞれ対応した、記憶領域206_{−1}〜206_{−64}, 206_{−65}〜206_{−128}, 206_{−129}〜206_{−192}, 206_{−193}〜206_{−256}, 206_{−257}〜206_{−320}, 206_{−321}〜206_{−384}, 206_{−385}〜206_{−448}, 206_{−449}〜206_{−512} (N=64)である。

【0188】また、上述したように、64個の記憶領域206_{−1}〜206_{−64}は、それぞれ参照フレームの横一行の参照ブロック数に対応した個数のブロック記憶領域BMを有している。さらに、各ブロック記憶領域BMには、参照ブロックを構成する画素データの個数分、本実施の形態では64個の計数値記憶領域を有している。

【0189】図31は、上述した64個の記憶領域206_{−1}〜206_{−64}をさらに詳細に示したものである。各メモリ部250_{−1}〜250_{−8}の8個の記憶領域(例えば、206_{−1}〜206_{−8})は、それぞれ、ロウ方向に64m個(mは横一行の参照ブロック数)、カラム方向に8n(nは計数値生成部205からの計数値の出力ビット数)のメモリセルがマトリクス状に配置されて構成されている。

【0190】上述したように、ある参照ブロック(参照ブロックP)に関する差分絶対値演算は1ライン目〜15ライン目の画素データを用いて行われる。例えば、1ライン目の画素データを用いて演算が行われるときは、記憶領域206_{−1}, 206_{−2}, 206_{−3}, 206_{−4}, 206_{−5}, 206_{−6}, 206_{−7}, 206_{−8}の記憶位置に書き込みが行われる。1ライン目で4画素目の画素データを用いた演算が行われるとき、演算器(7, 7)〜(4, 7)で行われる演算は参照ブロックPに関する演算であり、演算器(3, 7)〜(0, 7)で行われる演算は参照ブロックP−1に関する演算である。そのため、記憶領域206_{−1}, 206_{−2}, 206_{−3}, 206_{−4}, 206_{−5}, 206_{−6}, 206_{−7}, 206_{−8}の、例えば図31(1ライン目)に格子線で示した記憶位置に計数値の書き込みが行われる。

【0191】また例えば8ライン目の画素データを用いて演算が行われるときは、記憶領域206_{−1}〜206_{−8}の記憶位置に書き込みが行われる。8ライン目で6画素目の画素データを用いた演算が行われるとき、演算器(7, 7)〜(2, 7), (7, 6)〜(2, 6), (7, 5)〜(2, 5), (7, 4)〜(2, 4), (7, 3)〜(2, 3), (7, 2)〜(2, 2), (7, 1)〜(2, 1), (7, 0)〜(2, 0)で行われる演算は参照ブロックPに関する演算であり、演算器(1, 7)〜(0, 7), (1, 6)〜(0, 6), (1, 5)〜(0, 5), (1, 4)〜(0, 4), (1, 3)〜(0, 3), (1, 2)〜(0, 2),

(1, 1) ~ (0, 1)、(1, 0) ~ (0, 0)で行われる演算は参照ブロックP-1に関する演算である。そのため、記憶領域206₋₁ ~ 206₋₄の、例えば図31(8ライン目)に格子縞で示した記憶位置に計数値の書き込みが行われる。

【0192】次に、メモリ部250₋₁ ~ 250₋₈における書き込み用のメインロウデコーダ253およびサブロウデコーダ254についてさらに説明する。上述したように、メモリ部250₋₁においては、カラム方向に並ぶ複数のメモリセルを、差分絶対値演算器204₋₁ ~ 204₋₈((0, 0) ~ (7, 0))のそれぞれに対応したn個毎のセクションを独立してアクセスする必要がある。そのため、メモリ部250₋₁においては、メインロウデコーダ253によりアクセスすべきロウ位置が指定され、サブロウデコーダ254によりその指定されたロウ位置にある8個のセクションのうちアクセスすべきセクションが指定される。

【0193】また、上述したように、メモリ部250₋₁においては、2個の参照ブロックに対応したブロック記憶領域BMに存在する2つのロウ位置を同時にアクセスする必要がある。またその場合、2つのロウ位置では、カラム方向に互いに異なるセクションをアクセスする必要がある。そのため、メモリ部250₋₁においては、メインロウデコーダ253によりアクセスすべき2つのロウ位置が同時に指定され、サブロウデコーダ254によりその指定された2つのロウ位置でカラム方向に互いに異なるセクションが指定される。

【0194】以上のことは、メモリ部250₋₂ ~ 250₋₈においても同様である。以下、メモリ部250₋₁を例にとり、図32を参照して、メインロウデコーダ253およびサブロウデコーダ254、さらにはワード線構成の具体例を説明する。ここでは、参照ブロックPに対応した記憶領域およびこれに隣接した参照ブロックP+1に対応した記憶領域を中心に説明するが、その他の参照ブロックに対応した記憶領域はそれらの記憶領域を繰り返した構成となっている。

【0195】図において、カラム方向に連続したグローバルワード線WL₁, WL_{1,1}, WL_{1,2}, WL_{1,3}, ...は参照ブロックPに対応した記憶領域に配設されたものである。グローバルワード線WL₁, WL_{1,1}, WL_{1,2}, WL_{1,3}, ...のそれぞれに対応して、セクション毎に分割された分割ワード線としてのセクションワード線WL_{S0}, WL_{S1}, WL_{S2}, ..., WL_{S7}(WL_{S0}, WL_{S1}, WL_{S2}のみ図示)が配設されている。

【0196】また、カラム方向に連続したグローバルワード線WL_{1,65}, WL_{1,66}, WL_{1,67}, ...は参照ブロックP+1に対応した記憶領域に配設されたものである。グローバルワード線WL_{1,65}, WL_{1,66}, WL_{1,67}, ...のそれぞれに対応して、セクション毎に分割された分割ワード線としての

セクションワード線WL_{S0}, WL_{S1}, WL_{S2}, ..., WL_{S7}(WL_{S0}, WL_{S1}, WL_{S2}のみ図示)が配設されている。

【0197】ここで、1本のセクションワード線は、上述したあるブロック記憶領域BMに存在する1個の計数値記憶領域に対応している。この1本のセクションワード線には、カラム方向に並ぶn個のメモリセル131が接続されている。また、セクションワード線WL_{S0}, WL_{S1}, WL_{S2}, ..., WL_{S7}の部分は、それぞれ記憶領域206₋₁, 206₋₂, 206₋₃, ..., 206₋₈に対応している。

【0198】また、メインロウデコーダ253は、ロウデコーダRDECと、各グローバルワード線に対応して配設されたオア(OR)ゲートおよびノア(NOR)ゲートからなっている。

【0199】参照ブロックPに対応した記憶領域のグローバルワード線WL₁, WL_{1,1}, WL_{1,2}, WL_{1,3}, ...に対応してオアゲートGT₁, GT_{1,1}, GT_{1,2}, GT_{1,3}, ...が配設されている。これらオアゲートGT₁, GT_{1,1}, GT_{1,2}, GT_{1,3}, ...の出力側は、それぞれグローバルワード線WL₁, WL_{1,1}, WL_{1,2}, WL_{1,3}, ...に接続されている。

【0200】参照ブロックP+1に対応した記憶領域のグローバルワード線WL_{1,65}, WL_{1,66}, WL_{1,67}, ...に対応してノアゲートGT_{1,65}, GT_{1,66}, GT_{1,67}, ...が配設されている。これらノアゲートGT_{1,65}, GT_{1,66}, GT_{1,67}, ...の出力側は、それぞれグローバルワード線WL_{1,65}, WL_{1,66}, WL_{1,67}, ...に接続されている。

【0201】ロウデコーダRDECには、各グローバルワード線に対応した信号出力端子、反転信号出力端子を備えている。グローバルワード線WL₁, WL_{1,1}, WL_{1,2}, WL_{1,3}, ...に対応した信号出力端子に出力される信号("1"または"0")は、それぞれオアゲートGT₁, GT_{1,1}, GT_{1,2}, GT_{1,3}, ...の一方の入力側に供給される。これらオアゲートGT₁, GT_{1,1}, GT_{1,2}, GT_{1,3}, ...の他方の入力側には、それぞれ参照ブロックP-1に対応した記憶領域のグローバルワード線に対応したロウデコーダRDECの反転信号出力端子に出力される信号が反転されて供給される。

【0202】また、グローバルワード線WL_{1,65}, WL_{1,66}, WL_{1,67}, ...に対応した反転信号出力端子に出力される信号("1"または"0")は、それぞれノアゲートGT_{1,65}, GT_{1,66}, GT_{1,67}, ...の一方の入力側に反転されて供給される。これらノアゲートGT_{1,65}, GT_{1,66}, GT_{1,67}, ...の他方の入力側には、それぞれ参照ブロックPに対応した記憶領域のグローバルワード線WL

、 $WL_{1,1}, WL_{1,2}, WL_{1,3}, \dots$ に対応したロウデコーダRDECの信号出力端子に出力される信号がそのまま供給される。

【0203】また、サブロウデコーダ254は、各セクションワード線に対応して配設されたアンド(AND)ゲートからなっている。各アンドゲートの出力側はそれぞれ対応するセクションワード線に接続されている。参照ブロックPに対応した記憶領域の各セクションワード線に対応しては、それぞれアンドゲート AG_p が配設されている。各アンドゲート AG_p の一方の入力側には、対応するグローバルワード線に得られる信号(“1”または“0”)が供給される。参照ブロックP+1に対応した記憶領域の各セクションワード線に対応しては、それぞれアンドゲート AG_{p+1} が配設されている。各アンドゲート AG_{p+1} の一方の入力側には、対応するグローバルワード線に得られる信号(“1”または“0”)が反転されて供給される。

【0204】また、ロウ方向に並ぶアンドゲート AG_p, AG_{p+1} には、バッファBF₀, BF₁, BF₂, ..., BF_i(BF₀, BF₁, BF₂のみ図示)を介して共通の制御信号(“1”または“0”)が供給される。この場合、この制御信号は、アンドゲート AG_p にはそのまま供給されるが、アンドゲート AG_{p+1} には反転して供給される。

【0205】以上の構成において、参照ブロックP、P+1に対応した記憶領域の所定のグローバルワード線、例えば $WL_{1,1}, WL_{1,2}, \dots$ を活性化し、そしてグローバルワード線 $WL_{1,2}$ に対応したセクションワード線 WL_{s0}, WL_{s1} を活性化すると共に、グローバルワード線 $WL_{1,2}$ に対応したセクションワード線 $WL_{s2} \sim WL_{s7}$ を活性化する場合の動作について説明する。

【0206】この場合、ロウデコーダRDECは、グローバルワード線 $WL_{1,1}, WL_{1,2}, WL_{1,3}, WL_{1,4}, \dots$ に対応した信号出力端子のうち、グローバルワード線 $WL_{1,2}$ に対応した信号出力端子には“1”を出力し、その他の信号出力端子には“0”を出力する。また、ロウデコーダRDECは、グローバルワード線 $WL_{1,4}, WL_{1,5}, WL_{1,6}, WL_{1,7}, \dots$ に対応した信号出力端子の全てに“1”を出力する。またこの場合、参照ブロックP-1に対応した記憶領域のグローバルワード線に対応した信号出力端子の全てに“1”を出力する。

【0207】これにより、オアゲートGT_{1,2}の出力側に“1”が得られ、このオアゲートGT_{1,2}の出力側に接続されているグローバルワード線 $WL_{1,2}$ は活性化される。また同時に、ノアゲートのGT_{1,4}}の出力側に“0”が得られ、このノアゲートGT_{1,4}}の出力側に接続されているグローバルワード線 $WL_{1,4}$ は活性化される。

【0208】また、セクションワード線 WL_{s0}, WL_{s1} 50

に対応して、それぞれロウ方向に並ぶアンドゲート AG_p, AG_{p+1} にそれぞれ制御信号として“1”を入力する。また、セクションワード線 $WL_{s2} \sim WL_{s7}$ に対応して、それぞれロウ方向に並ぶアンドゲート AG_p, AG_{p+1} にそれぞれ制御信号として“0”を入力する。

【0209】これにより、グローバルワード線 $WL_{1,2}$ に対応した8本のセクションワード線のうち、セクションワード線 WL_{s0}, WL_{s1} のみ“1”が印加された状態となって活性化される。また同時に、グローバルワード線 $WL_{1,4}$ に対応した8本のセクションワード線のうち、セクションワード線 $WL_{s2} \sim WL_{s7}$ のみ“1”が印加された状態となって活性化される。

【0210】以上は、参照ブロックP、P+1に対応した記憶領域の2つのロウ位置にあるグローバルワード線(セクションワード線)を同時に活性化する動作を説明したが、その他の2つの参照ブロックに対応した記憶領域の2つのロウ位置にあるグローバルワード線(セクションワード線)を同時に活性化する場合にも同様の動作によって行うことができる。

【0211】例えば、参照ブロックP+1、P+2に対応した記憶領域の2つのロウ位置にあるグローバルワード線(セクションワード線)を同時に活性化する場合、ロウデコーダRDECは、参照ブロックP+1に対応した記憶領域の8本のグローバルワード線のうち、活性化すべきグローバルワード線に対応した反転信号出力端子に“0”を出力し、その他の反転信号出力端子に“1”を出力する。また、ロウデコーダRDECは、参照ブロックP+2の記憶領域の8本のグローバルワード線に対応した信号出力端子の全てに“0”を出力する。また、この場合、ロウデコーダRDECは、参照ブロックPの記憶領域の8本のグローバルワード線に対応した信号出力端子の全てに“0”を出力する。

【0212】次に、複数の加算器209₁～209_nおよびヒストグラムテーブル生成用の半導体メモリ210の詳細を説明する。複数の加算器209₁～209_nと半導体メモリ210は一体化されており、複数の加算器209₁～209_nを構成するそれぞれのビット単位の複数の加算部は、半導体メモリ210のカラムのピッチに揃えて配されている。

【0213】図33は、加算器209₁およびそれに対応した半導体メモリ210の記憶領域210₁の部分の詳細構成を示したものである。なお、説明は省略するが、加算器209₂～209_nおよびそれに対応した半導体メモリ210の記憶領域210₂～210_nの部分についても同様に構成されている。

【0214】図33において、記憶領域210₁には、カラム方向にn個、ロウ方向に1個のメモリセル(Memory Cell)130が配されている。この場合、カラム方向に並ぶn個のメモリセル130によって、1つの度数記憶領域が構成されている。メモリセル130は、上述の

図9で説明したように、書き込みおよび読み出し用の第1のポートと読み出し専用の第2のポートとを有する2ポート構成のものである。

【0215】カラム方向に並ぶメモリセル130に沿って、ワード線WL、および読み出し専用ワード線WRLが配されている。ワード線WLはメモリセル130の端子13に接続され、読み出し専用ワード線WRLはメモリセル130の端子17に接続される。

【0216】また、カラム方向に並ぶn個のメモリセル130に、それぞれビット線BL₀〜BL_{n-1}、/BL₀〜/BL_{n-1}、および読み出し専用ビット線BRL₀〜BRL_{n-1}が接続されている。上述したように、ビット線BL₀〜BL_{n-1}はメモリセル130の端子14に接続され、ビット線/BL₀〜/BL_{n-1}はメモリセル130の端子15に接続され、読み出し専用ビット線BRL₀〜BRL_{n-1}はメモリセル130の端子16に接続される。

【0217】この図33におけるその他の構成は、図8における加算器125₋₁および記憶領域126₋₁の構成と同様とされる。図33に示す加算器209₋₁および記憶領域210₋₁の部分の動作を説明する。まず、カラム方向に並ぶn個のメモリセル130によって1個の度数記憶領域が構成されているが、このn個のメモリセル130の記憶データをクリアする動作について説明する。

【0218】度数記憶領域を構成するn個のメモリセル130の記憶データをクリアする場合、書き込み制御信号φ_wおよびクリア制御信号φ_{clr}はアクティブ、つまり“1”とされ、読み出し制御信号φ_rおよびイコライズ制御信号φ_{eq}はインアクティブ、つまり“0”とされ、さらにワード線WLがワード線が活性化される。

【0219】この場合、クリア制御信号φ_{clr}がアクティブとされてトランジスタQ51がオンとなる。そのため、“0”のデータが生成され、このデータがビット線BL₀〜BL_{n-1}に出力される。したがって、ワード線WLを活性化することで、度数記憶領域を構成するn個のメモリセル130には“0”のデータが書き込まれ、記憶データのクリアが行われる。

【0220】次に、度数記憶領域(n個のメモリセル130)に記憶されている記憶データMD₀〜MD_{n-1}に、8ビットの計数値D₀〜D₇を、加算器209₋₁(加算部140₁〜140_{n-1})で足し込み、そして加算器209₋₁で得られた足し込みデータAD₀〜AD_{n-1}を、当該度数記憶領域に書き戻す動作について説明する。

【0221】所定の度数記憶領域に記憶されている記憶データMD₀〜MD_{n-1}に、8ビットの計数値D₀〜D₇を足し込む場合、最初に、イコライズ制御信号φ_{eq}はアクティブ、つまり“1”とされ、書き込み制御信号φ_w、読み出し制御信号φ_rおよびクリア制御信号φ_{clr}はインアクティブ、つまり“0”とされ、ビット線対BL₀、/BL₀〜BL_{n-1}、/BL_{n-1}のイコライズ(プリチャージ)が行われる。

【0222】この場合、ビット線対BL₀、/BL₀に関しては、イコライズ制御信号φ_{eq}がアクティブとされてトランジスタQ31〜Q33の全てがオンとなり、ビット線BL₀およびビット線/BL₀に電源の電位が印加され、これらビット線BL₀およびビット線/BL₀は同電位となる。他のビット線対BL₁、/BL₁〜BL_{n-1}、/BL_{n-1}に関しても同様である。

【0223】このようにビット線対BL₀、/BL₀〜BL_{n-1}、/BL_{n-1}のイコライズが行われた状態で、読み出し制御信号φ_rはアクティブ、つまり“1”とされ、書き込み制御信号φ_w、イコライズ制御信号φ_{eq}およびクリア制御信号φ_{clr}はインアクティブ、つまり“0”とされ、さらにワード線WLが活性化される。

【0224】これにより、度数記憶領域を構成するn個のメモリセル130の記憶データMD₀〜MD_{n-1}が、それぞれビット線対BL₀、/BL₀〜BL_{n-1}、/BL_{n-1}およびセンスアンプSA₀〜SA_{n-1}を介して読み出され、加算部140₀〜140_{n-1}のB側の入力端子にそれぞれ入力される。したがって、度数記憶領域に記憶されている記憶データMD₀〜MD_{n-1}に、8ビットの計数値D₀〜D₇が足し込まれる。

【0225】そして、加算部140₀〜140_{n-1}における加算出力、つまり足し込みデータAD₀〜AD_{n-1}が有効になったところで、書き込み制御信号φ_wはアクティブ、つまり“1”とされ、読み出し制御信号φ_r、イコライズ制御信号φ_{eq}およびクリア制御信号φ_{clr}はインアクティブ、つまり“0”とされ、さらにワード線WLが活性化される。

【0226】この場合、加算部140₀の部分に関して、足し込みデータS₀が“1”である場合には、トランジスタQ11はオン、トランジスタQ12はオフとなり、ビット線/BL₀に“0”が出力されることから、度数記憶領域を構成するn個のメモリセル130のうち、当該加算部140₀に対応するメモリセル130には、“1”のデータが記憶される。一方、加算部140₀の部分に関して、足し込みデータS₀が“0”である場合には、トランジスタQ11はオフ、トランジスタQ12はオンとなり、ビット線BL₀に“0”が出力されることから、度数記憶領域を構成するn個のメモリセル130のうち、当該加算部140₀に対応するメモリセル130には、“0”のデータが記憶される。

【0227】他の加算部140₁〜140_{n-1}の部分に関しても同様である。これにより、加算器209₋₁で得られた足し込みデータAD₀〜AD_{n-1}は、度数記憶領域を構成するn個のメモリセル130に書き戻される。

【0228】なお、足し込みの動作において、オーバーフローとなる場合には、加算部140_{n-1}のキャリ出力端子C_{out}に得られるMSBのキャリ出力C_{msb}が“1”となるため、トランジスタQ13はオフとなり、足し込みデータAD₀〜AD_{n-1}が、度数記憶領域を構成するn個

のメモリセル130に書き込まれることはない。

【0229】その代わり、この場合、トランジスタQ61がオンとなる他に、トランジスタQ62もオンとなるので、ビット線/BL₀〜/BL_{n-1}にそれぞれ“0”の信号が出力される。したがって、度数記憶領域を構成するn個のメモリセル130のそれぞれに“1”のデータが書き込まれる。つまり、この度数記憶領域には最大値が記憶される。

【0230】次に、度数記憶領域に記憶された、度数（計数値）の累積値を、読み出す場合の動作を説明する。最初に、プリチャージ制御信号/ ϕ_{pc} がアクティブ、つまり“1”とされ、読み出し専用ビット線BRL₀〜BRL_{n-1}のプリチャージが行われる。この場合、トランジスタQ41はオンとなり、読み出し専用ビット線BRL₀〜BRL_{n-1}のそれぞれに電源の電位が印加される。

【0231】このように読み出し専用ビット線BRL₀〜BRL_{n-1}のプリチャージが行われた状態で、読み出し専用ワード線WRLが活性化される。これにより、度数記憶領域を構成するn個のメモリセル130の記憶データ Σ_0 〜 Σ_{n-1} が、それぞれ読み出し専用ビット線BRL₀〜BRL_{n-1}に得られる。ここで、記憶データ Σ_0 〜 Σ_{n-1} は、nビットの度数の累積値を構成している。

【0232】以上説明したように、図15に示す動きベクトル検出回路200においては、参照フレームの所定の参照ブロックを構成する画素データとこの画素データに対応した探索フレームの探索範囲に存在する複数の画素データとの間の差分絶対値演算を行い、参照ブロックを構成する各画素データ毎に、探索範囲の各画素位置に対応して、差分絶対値がその閾値より小さいときは計数値として0以外の値を配し、差分絶対値がその閾値以上のときは計数値として0を配した計数値テーブルを生成し、参照ブロックを構成する各画素データに対応した計数値テーブルの各計数値を度数とし、この度数を探索範囲の画素位置毎に足し込んでヒストグラムテーブルを生成し、このヒストグラムテーブルの度数の極大値に基づいて参照ブロックの動きベクトルMV₁〜MV_nを検出するものである。

【0233】この場合、このヒストグラムテーブルにおける度数の極大値は、参照ブロック内に存在する動きに対応した動きベクトルを表している。マッチング度合いが高く差分絶対値がその閾値より小さいときのみ計数値として0以外の値を配して計数値テーブルを生成するのであって、ヒストグラムテーブルはマッチング度合いが高いもののみによって生成されることとなり、その度数の分布はより先鋭化されたものとなる。そのため、参照ブロック内に動きの異なるものが複数混在する場合には、ヒストグラムテーブルに複数の極大値が明瞭に分離して示され易くなる。したがって、参照ブロックの一個または複数の動きベクトルMV₁〜MV_nを正しく検

出できる。

【0234】なお、差分絶対値がその閾値より小さいときに配する計数値を正の値としているが（表1参照）、これを負の値としてもよい。その場合、ヒストグラムテーブルには、参照ブロック内に存在する動きに対応した動きベクトルを表す度数の極小値が示される。この場合であっても、上述した実施の形態と同様に、ヒストグラムテーブルに基づいて、参照ブロックの動きベクトルを正しく検出できる。

10 【0235】また、マッチング度合いが高く差分絶対値がその閾値より小さいときに配する計数値を一定値とするのではなく、差分絶対値が小さい程、つまりマッチングの度合いが高い程、絶対値の大きな値としているので、ヒストグラムテーブルの度数の分布を一層先鋭化できる利益がある。

【0236】また、補正係数取得部207では、参照ブロックを構成する各画素データ毎にアクティビティAを求め、この各画素データ毎に、アクティビティAの最大アクティビティA_{max}に対する割合が高い程大きな値を補正係数kとして得て、参照ブロックを構成する各画素データに対応した計数値テーブルの計数値に、それぞれこの各画素データに対応した補正係数kを掛けて補正するものであり、アクティビティAによるマッチングの度合い、つまり差分絶対値の変化による不具合を軽減できる。

【0237】なお、補正係数取得部207で補正係数kを求め、計数値補正部208でこの補正係数kを計数値に掛けて補正するものであったが、この代わりに、計数値生成部205で差分絶対値に対応して計数値を生成する際に用いられる閾値を、アクティビティが大きい程大きくするように変更してもよい。この場合、計数値を補正係数kで補正する場合と同様に、周囲におけるアクティビティが大きな参照ブロックの画素データに対応した計数値テーブルの計数値に関しては、その絶対値が大きくなる。これにより、アクティビティによるマッチングの度合い、つまり差分絶対値の変化による不具合を軽減できる。

【0238】また、探索フレームとしての入力フレームを次フレームの参照フレームとして格納する参照フレームメモリ203を備え、複数の差分絶対値演算器204₁〜204_nで、入力フレームの画素データが入力される都度、その画素データと、フレームメモリ203より読み出される、その画素データを探索範囲に含む参照フレームの所定の参照ブロックの画素データとの差分絶対値を演算するものである。したがって、探索フレームを一旦フレームメモリに記憶し、その後改めて処理のために読み出すといった無駄な動作を排除でき、効率よく動きベクトルを検出できる。

【0239】また、半導体メモリ206のカラム方向に延びる各ワード線は、複数の記憶領域206₁〜206

にそれぞれ対応して分割された複数のセクションワード線からなり、半導体メモリ206は任意のワード線を構成する複数のセクションワード線のうち任意のセクションワード線を選択的に活性化させるためのロウデコード手段としてのメインロウデコード253およびサブロウデコード254を有するものであり、半導体メモリ206には、カラム方向にセクションワード線単位で独立してアクセスできる。

【0240】また、半導体メモリ206の複数の記憶領域206₋₁~206_{-n}は、それぞれ複数の参照ブロックに係る計数値を記憶するための複数のブロック記憶領域BMがロウ方向に配列されてなり、メインロウデコード253は2つの連続するブロック記憶領域BMに対応したグローバルワード線を活性化させ、サブロウデコード254はその2つのグローバルワード線に対応してそれぞれ活性化させるセクションワード線を互いに異なる記憶領域に対応したセクションワード線とするものである。

【0241】したがって、複数の差分絶対値演算器204₋₁~204_{-n}で2つの参照ブロックに係る差分絶対値演算が並行して行われても、計数値生成部205で生成された計数値を参照ブロック毎にロウ方向の異なるワード線位置に振り分けて書き込むことが可能となり、半導体メモリ206に参照ブロック別に、計数値テーブルを得ることができる。

【0242】また、加算器209₋₁~209_{-n}および半導体メモリ210が一体化され、加算器を構成するビット単位の複数の加算部140₀~140₋₁が、半導体メモリ210のカラムのピッチに揃えて配されるものである(図33参照)。したがって、加算器209₋₁~209_{-n}から半導体メモリ210への足し込みデータの供給および半導体メモリ210から加算器209₋₁~209_{-n}への記憶データの供給を効率的に行うことができる。

【0243】また、半導体メモリ210が、複数の加算器に関連して設けられた書き込みおよび読み出し用の第1のポートと、度数の累積値を読み出すための読み出し専用の第2のポートとを有するものである(図46参照)。したがって、半導体メモリ210の記憶データの読み出しを、加算器209₋₁~209_{-n}による演算とは別個独立して行うことができる。

【0244】また、半導体メモリ210の度数記憶領域を構成するメモリセル130の記憶データをクリアする際に、トランジスタQ51をオンとして、クリアするための“0”データを生成し、このデータをメモリセル130に書き込みデータとして供給するものであり、外部からクリアするためのデータを入力することなく、半導体メモリ210を容易にクリアすることができる。

【0245】また、加算器209₋₁~209_{-n}をそれぞれ構成する複数の加算部140₀~140₋₁による演算結果がオーバフローとなる場合、複数の加算部140₀~140₋₁に対応した半導体メモリ210の度数記憶領域に最大値を記憶(セット)するものであり、この度数記憶領域に誤った小さな値が度数累積値として記憶され、動きベクトルの検出誤りが生じることを防止できる。

【0246】また、半導体メモリ210の記憶領域210₋₁~210_{-n}では、ビット線BL₀~BL₋₁をトランジスタQ51を介して接地し、このトランジスタQ51のゲートにクリア制御信号/φ_{cl}を入力する構成とし、クリア信号φ_{cl}をアクティブとしたとき、度数記憶領域を構成するn個のメモリセル130に“0”のデータが書き込まれてクリアされるものを示した(図33参照)。

【0247】ここで、図33に破線図示するように、ビット線/BL₀~/BL₋₁をトランジスタQ52を介して接地し、このトランジスタQ52のゲートにクリア制御信号/φ_{cl}を入力する構成とした場合、クリア信号φ_{cl}をアクティブとしたとき、トランジスタQ52で“1”のデータが生成され、所定の記憶部を構成するn個のメモリセル130に“1”のデータが書き込まれる。

【0248】したがって、ビット線対BL₀、/BL₀~BL₋₁、/BL₋₁のそれぞれに対応してトランジスタQ51、Q52を設けておき、それらのいずれかを選択的にビット線に接続し、クリア信号φ_{cl}をアクティブとしたとき度数記憶領域を構成するn個のメモリセル130に、所定のデータをプリセットするようにしてもよい。このプリセットデータを工夫することで、例えば平坦な絵柄の部分においては(0,0)等の特定の動きベクトルが検出されやすくなる。このプリセットの設定は、半導体装置の設計時に予め決めておけばよく、従ってコンタクトレイヤのプログラムなどによって設定することが想定される。

【0249】また、加算器209₋₁~209_{-n}およびヒストグラムテーブル生成用の半導体メモリ210とを一体化したものであるが、さらに計数値補正部208、動きベクトル検出部211等をも一体化するようにしてもよい。

【0250】また、動きベクトル検出部211で検出された参照ブロックの動きベクトルMV₁~MV_nに基づいて、動きベクトル特定部212で、その参照ブロックを構成する各画素データの動きベクトルmvを特定することができる。上述したように、参照ブロックの動きベクトルを正しく検出できることから、その動きベクトルに基づいて特定される各画素データの動きベクトルもより精度の高いものとなる。

【0251】さらに、動きベクトル特定部212では、このように特定される参照ブロックを構成する各画素データの動きベクトルmvから孤立した動きベクトルを除去する等の補正処理を行うので、この各画素データの動きベクトルMV₁~MV_nに基づいて、動きベクトル特定部212で、その参照ブロックを構成する各画素データの動きベクトルmvを特定することができる。上述したように、参照ブロックの動きベクトルを正しく検出できることから、その動きベクトルに基づいて特定される各画素データの動きベクトルもより精度の高いものとなる。

【0251】さらに、動きベクトル特定部212では、このように特定される参照ブロックを構成する各画素データの動きベクトルmvから孤立した動きベクトルを除去する等の補正処理を行うので、この各画素データの動きベクトルMV₁~MV_nに基づいて、動きベクトル特定部212で、その参照ブロックを構成する各画素データの動きベクトルmvを特定することができる。上述したように、参照ブロックの動きベクトルを正しく検出できることから、その動きベクトルに基づいて特定される各画素データの動きベクトルもより精度の高いものとなる。

きベクトルの精度をさらに高めることができる。

【0252】

【発明の効果】この発明によれば、演算器およびメモリが一体化され、演算器を構成するビット単位の複数の演算部がメモリのカラムのピッチに揃えて配されるものであり、演算器からメモリへの演算データの供給および半導体メモリから演算器への記憶データの供給を効率的に行うことができる。

【0253】また、この発明によれば、演算器は加算器であり、入力データとメモリの記憶データとを複数の演算部で加算し、この加算データをメモリに記憶データとして記憶するものであり、メモリの記憶データに入力データを順次足し込むことを良好に行うことができる。

【0254】また、この発明によれば、メモリが、複数の演算部に関連して設けられた書き込みおよび読み出し用の第1のポートと読み出し専用の第2のポートとを有するものであり、メモリの記憶データの読み出しを、演算器による演算とは別途独立して行うことができる。

【0255】また、この発明によれば、メモリをクリアまたはプリセットするためのデータを生成し、このデータによってメモリをクリアまたはプリセットするものであり、外部からクリアまたはプリセットするためのデータを入力することなく、メモリを容易にクリアまたはプリセットすることができる。

【0256】また、この発明によれば、演算器を構成する複数の演算部による演算結果がオーバーフローとなる場合、複数の演算部に対応したメモリの所定領域に最大値をセットするものであり、この半導体メモリの所定領域に演算結果として誤った小さな値が記憶されることを防止できる。

【図面の簡単な説明】

【図1】実施の形態としての動き補償予測符号化装置の構成を示すブロック図である。

【図2】動き検出のためのブロックマッチング法を説明するための図である。

【図3】動き検出のためのブロックマッチング法を説明するための図である。

【図4】動き検出のためのブロックマッチング法を説明するための図である。

【図5】動き検出のためのブロックマッチング法を説明するための図である。

【図6】動きベクトル検出回路の構成を示すブロック図である。

【図7】1個の参照画素とN個の探索範囲画素との1対Nのマッチング演算を説明するための図である。

【図8】半導体メモリと加算器とを一体化した構成を示す図である。

【図9】メモリセルの構成を示す図である。

【図10】ブロックマッチング法の問題点（複数の動きがある場合）を説明するための図である。

【図11】動きベクトル検出方法の概略を説明するための図である。

【図12】相関値テーブルの一例（2つの動きがある場合）を示す図である。

【図13】ヒストグラムテーブルの一例（2つの動きがある場合）を示す図である。

【図14】アクティビティを求めるための画素データ例を示す図である。

【図15】動きベクトル検出回路の構成を示すブロック図である。

【図16】入力フレームが探索フレームであって過去フレームが参照フレームである場合における、動きベクトル検出のためのマッチング演算を説明するための図である。

【図17】参照ブロックと探索範囲との関係を示す図である。

【図18】動きベクトル特定部の構成を示すブロック図である。

【図19】入力画素データと参照ブロックと演算器との関係を示す図である。

【図20】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図21】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図22】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図23】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図24】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図25】入力画素データの演算対象範囲と参照ブロックとの位置関係を示す図である。

【図26】所定の参照ブロックを構成する各画素データに対応した計数値テーブルを得る際の、複数の差分絶対値演算器における演算過程を示す図である。

【図27】演算器（7，7）における演算過程を示す図である。

【図28】動きベクトル（1参照画素の探索範囲の座標）を示す図である。

【図29】動きベクトルと計数値テーブル生成用メモリにおける記憶位置との対応関係を示す図である。

【図30】半導体メモリの全体構成を示す図である。

【図31】半導体メモリの記憶領域の構成を示す図である。

【図32】メモリ部のメイン・サブrowデコーダとワード線の構成を示す図である。

【図33】半導体メモリと加算器とを一体化した構成を示す図である。

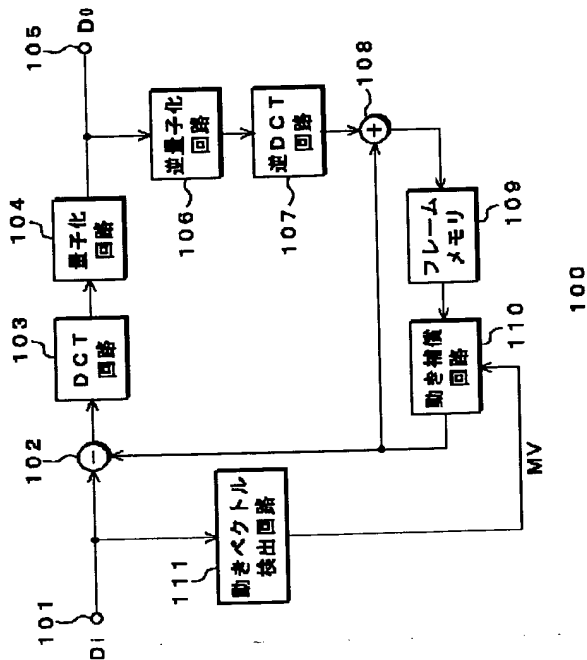
【符号の説明】

100・・・動き補償予測符号化装置、101・・・入

力端子、102・・・減算器、103・・・DCT回路、104・・・量子化回路、105・・・出力端子、106・・・逆量子化回路、107・・・逆DCT回路、108・・・加算器、109・・・フレームメモリ、110・・・動き補償回路、111、200・・・動きベクトル検出回路、121、201・・・システムコントローラ、122、202・・・入力端子、123・・・探索フレームメモリ、124₋₁～124_{-n}、204₋₁～204_{-n}・・・差分絶対値演算器、125₋₁～125_{-n}、209₋₁～209_{-n}・・・加算器、126・・・相関値テーブル用の半導体メモリ、126₋₁～126_{-n}、206₋₁～206_{-n}、210₋₁～210_{-n}・・・記憶領域、127・・・相関値テーブル評価器、128、213・・・出力端子、130・・・メモリセル、14*

【図1】

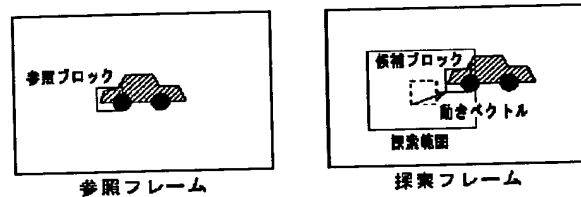
動き補償予測符号化装置



* 0₀～140_{n-1}・・・加算部、203・・・参照フレームメモリ、205・・・計数値生成部、206・・・計数値テーブル生成用の半導体メモリ、207・・・補正係数取得部、208・・・計数値補正部、210・・・ヒストグラムテーブル生成用の半導体メモリ、211・・・動きベクトル検出部、212・・・動きベクトル特定部、221・・・コントローラ、222・・・探索画素用のメモリ、223₋₁～223_{-n}・・・相関演算部、224・・・動きベクトル仮特定部、225・・・動きベクトル補正部、250₋₁～250_{-s}・・・メモリ部、251・・・読み出し用のロウデコーダ、253・・・書き込み用のメインロウデコーダ、254・・・書き込み用のサブロウデコーダ

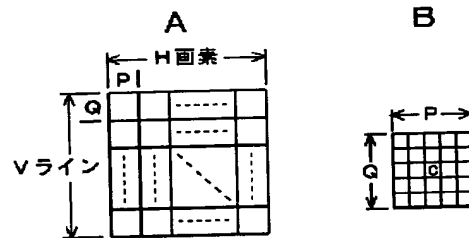
【図2】

ブロックマッチング法



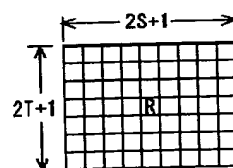
【図3】

ブロックマッチング法

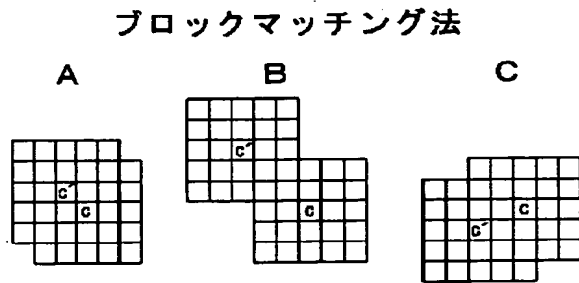


【図5】

ブロックマッチング法

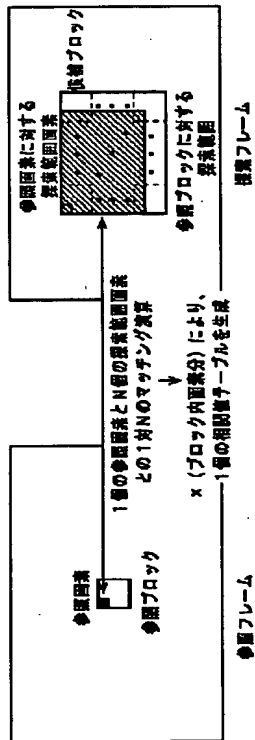


【図4】



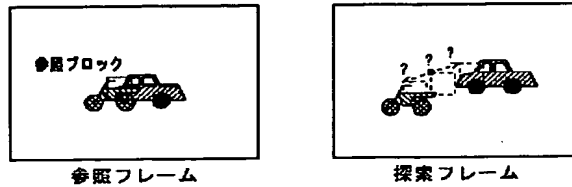
【図7】

マッチング演算

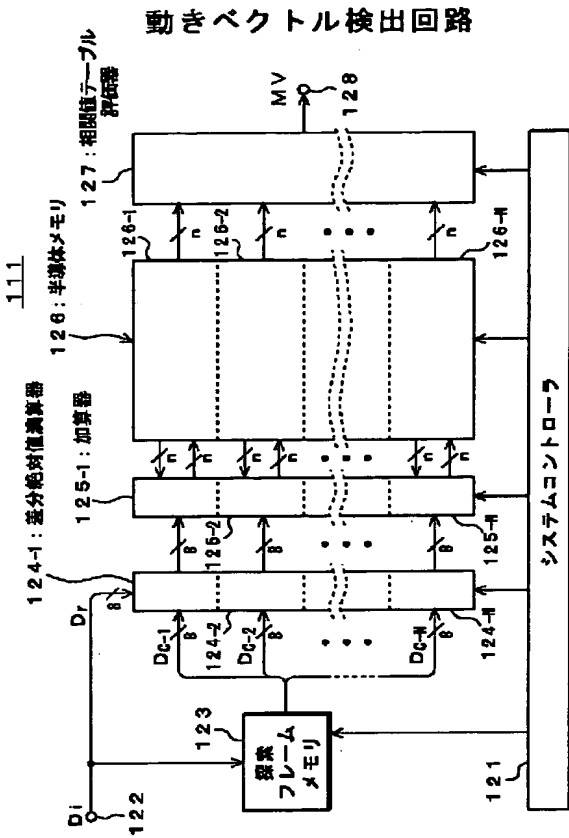


【図10】

ブロックマッチング法の問題点 (複数の動きがあるブロック)

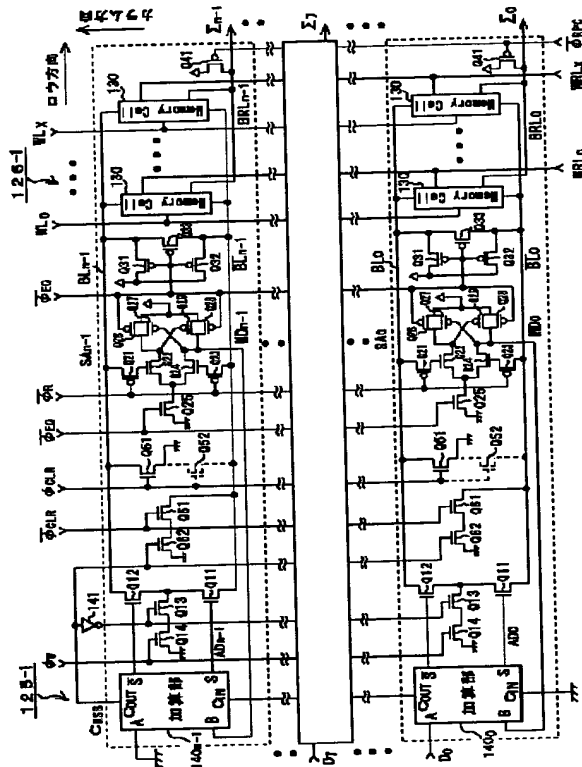


【図6】



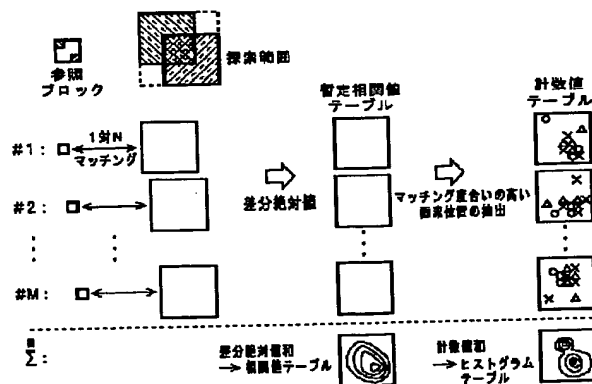
【図8】

半導体メモリと加算器とを一体化した構成



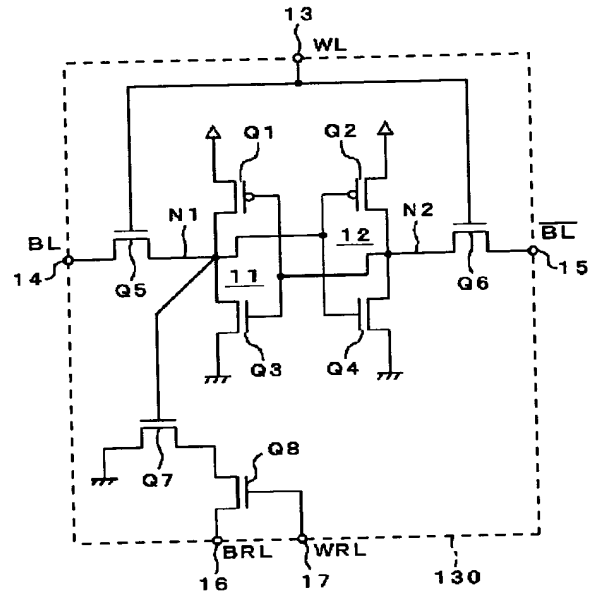
【図11】

動きベクトル検出方法の概略



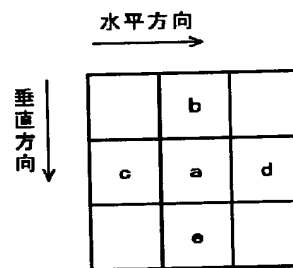
【図9】

メモリセル



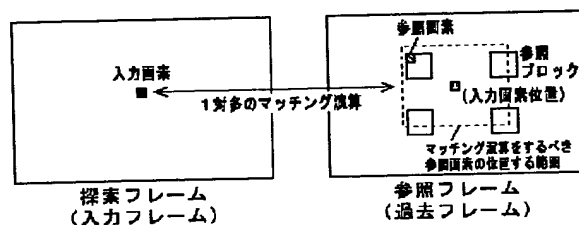
【図14】

アクティビティを求めるための画素データ例

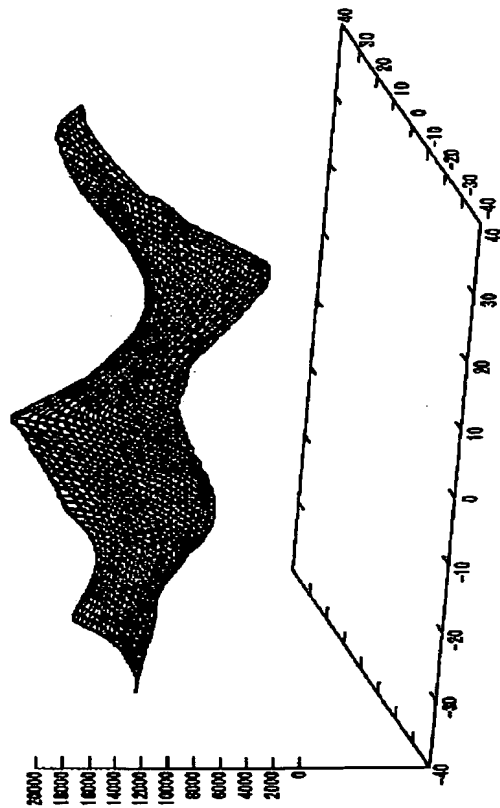


【図16】

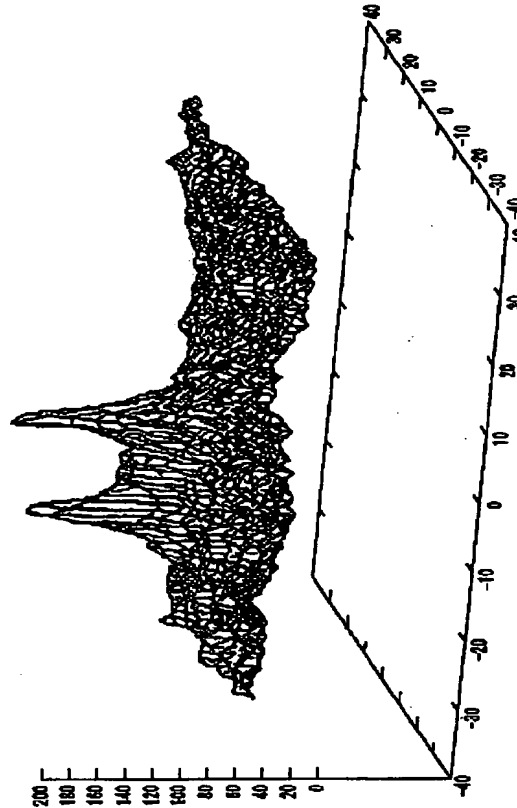
マッチング演算



【図12】

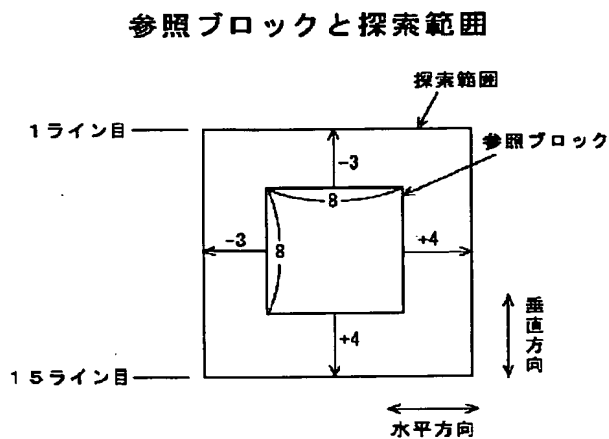


【図13】



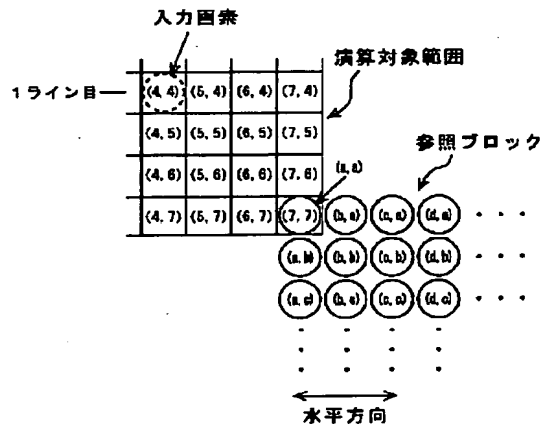
相関値テーブルの一例(2つの動きがある場合) ヒストグラムテーブルの一例(2つの動きがある場合)

【図17】

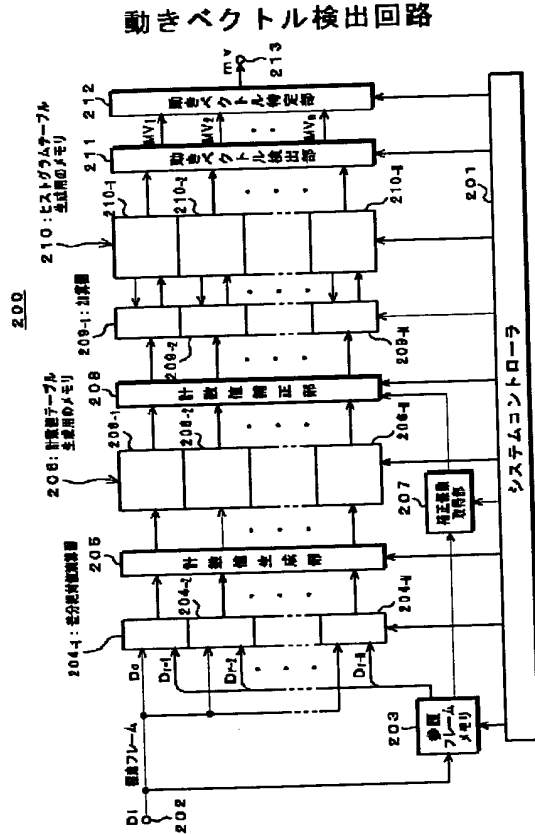


【図20】

入力画素の演算対象範囲と参照ブロックとの位置関係

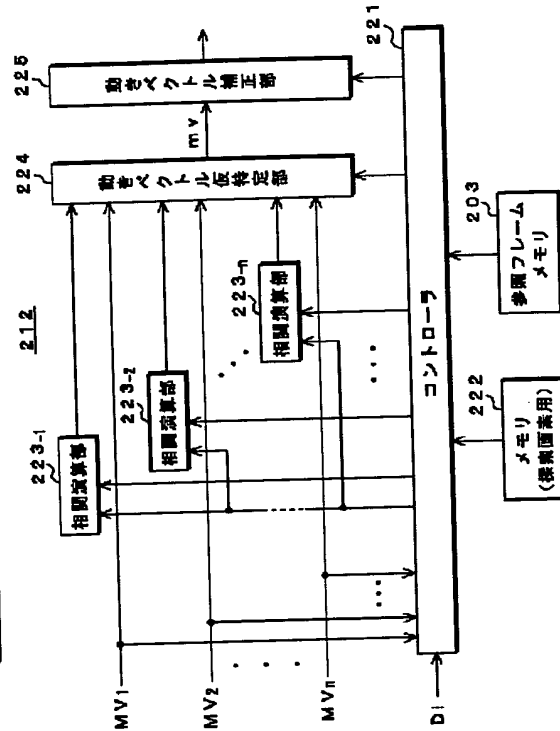


【図15】



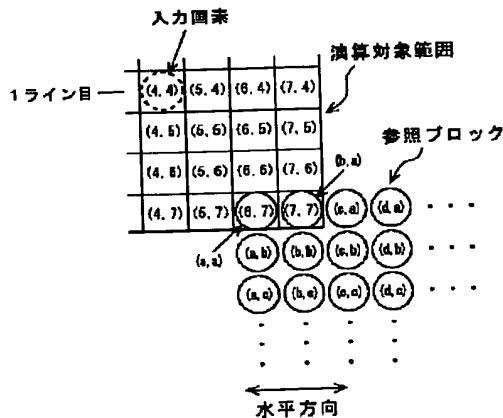
【図18】

動きベクトル特定部



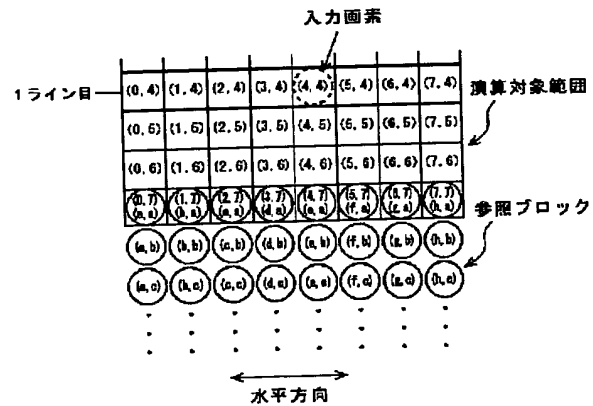
【図21】

入力画素の演算対象範囲と参照ブロックとの位置関係



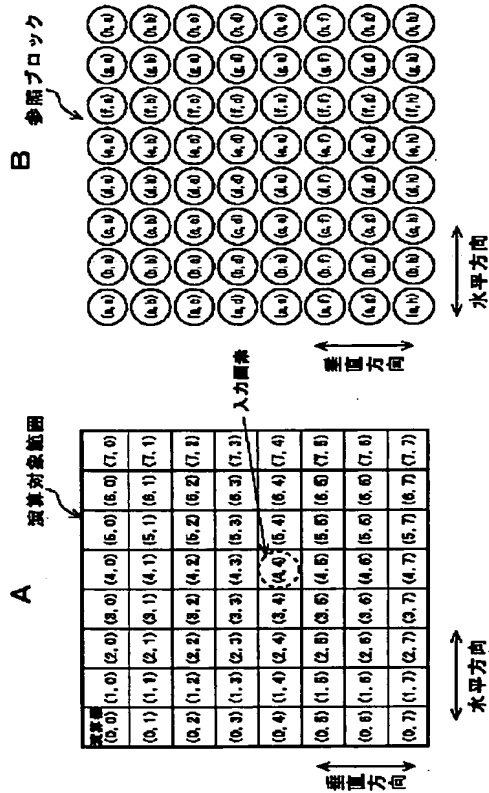
【図22】

入力画素の演算対象範囲と参照ブロックとの位置関係



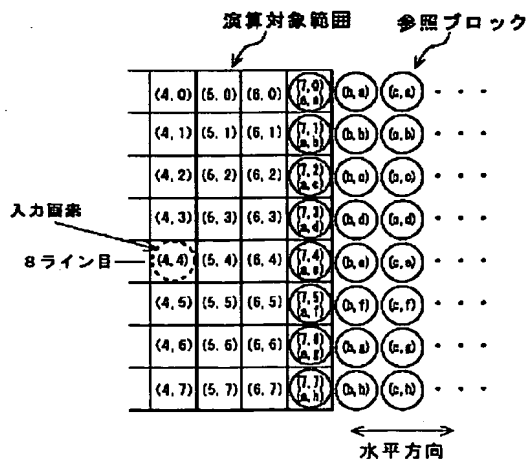
【図19】

入力画素と参照ブロックと演算器との関係



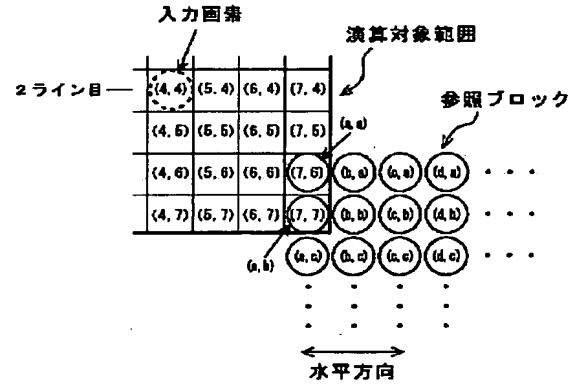
【図24】

入力画素の演算対象範囲と参照ブロックとの位置関係



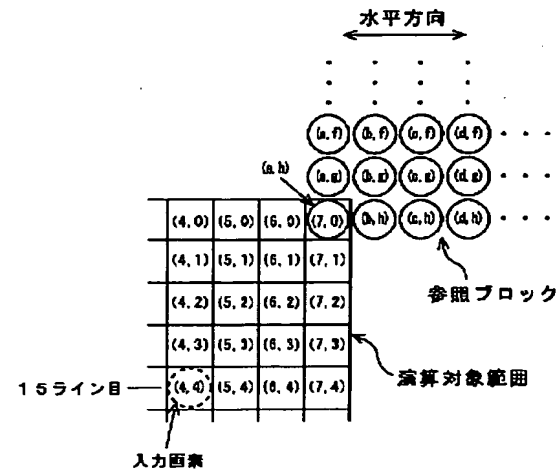
【図23】

入力画素の演算対象範囲と参照ブロックとの位置関係

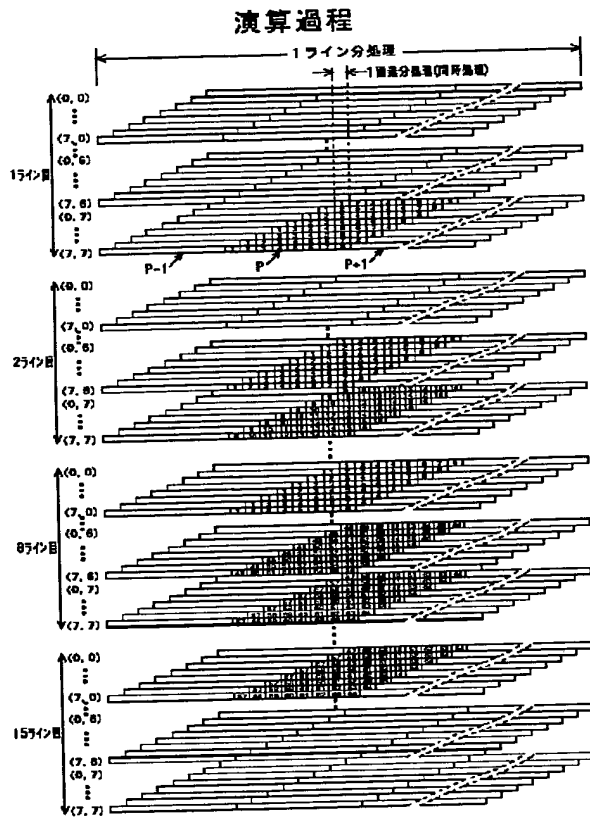


【図25】

入力画素の演算対象範囲と参照ブロックとの位置関係

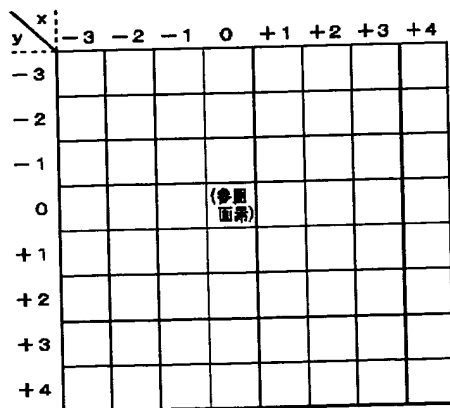


【図26】



【図28】

動きベクトル(探索範囲の座標)



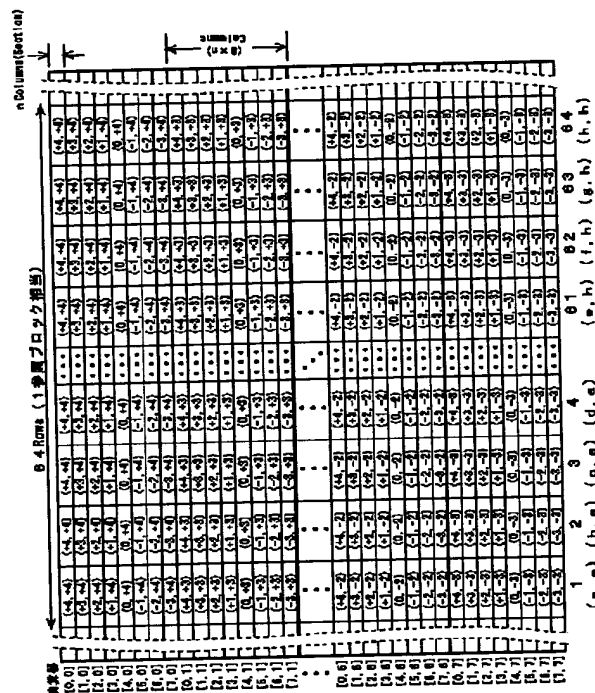
【図27】

演算器(7,7)における演算過程

1ライン目	1 (a, a)	2 (b, a)	3 (c, a)	4 (d, a)	5 (e, a)	6 (f, a)	7 (g, a)	8 (h, a)
2ライン目	9 (a, b)	10 (b, b)	11 (c, b)	12 (d, b)	13 (e, b)	14 (f, b)	15 (g, b)	16 (h, b)
8ライン目	57 (a, h)	58 (b, h)	59 (c, h)	60 (d, h)	61 (e, h)	62 (f, h)	63 (g, h)	64 (h, h)

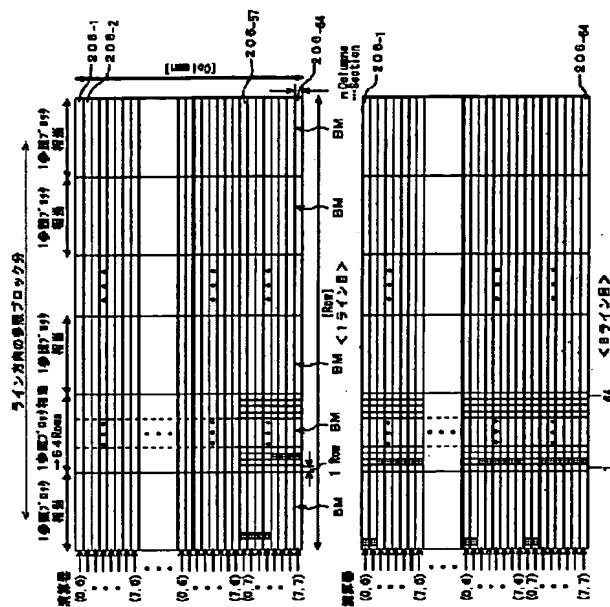
【図29】

動きベクトルと計数値テーブル生成用メモリの記憶位置との関係



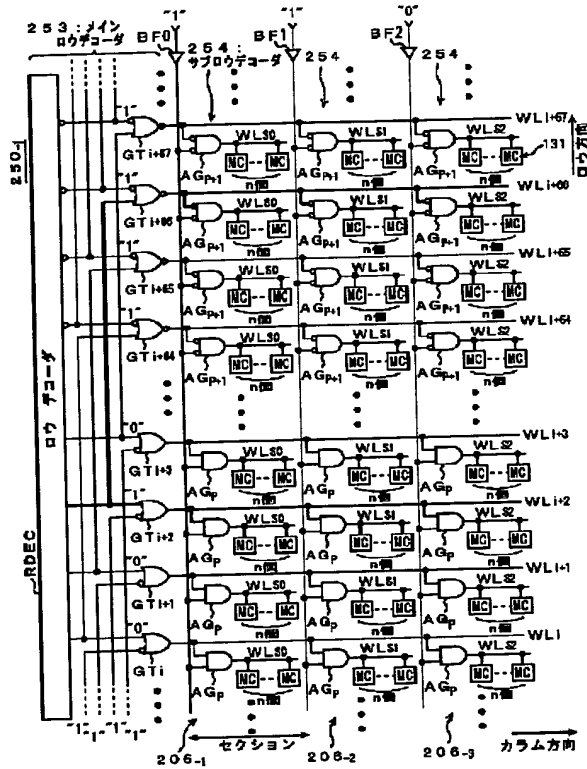
【図31】

半導体メモリの記憶領域の構成



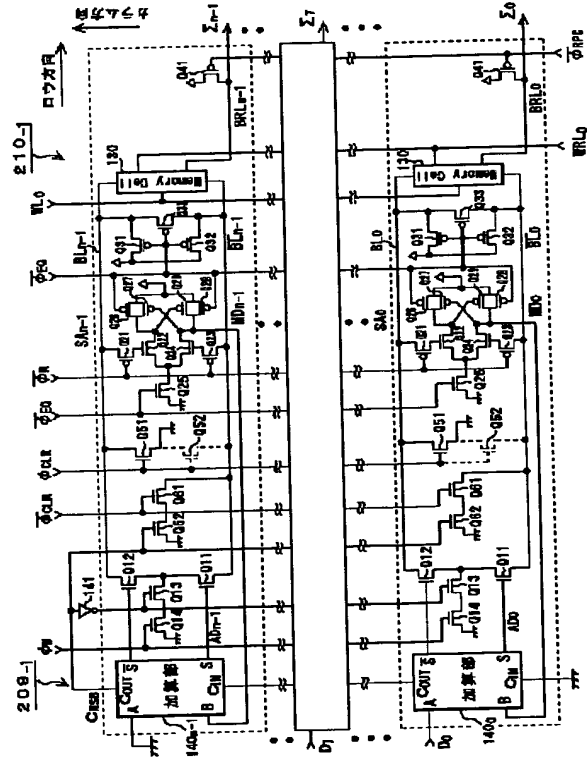
【図32】

メモリ部のメイン・サブロウデコーダとワード線



【図33】

半導体メモリと加算器とを一体化した構成



フロントページの続き

Fターム(参考) 5B015 HH03 JJ21 KB36 NN01 PP08
 5C059 KK13 KK19 KK50 MA23 MC11
 NN01 NN21 NN28 PP04 UA02
 UA33